I N T E R O P E R A B I L I D A D E E N T R E S I S T E M A S D E I N F O R M A Ç Ã O B A S E A D O S N A W E B

MESTRADO EM TECNOLOGIAS DAS ENGENHARIAS

Maria Judite Macedo Fernandes Saraiva

Universidade de Trás-os-Montes e Alto Douro

Departamento de Engenharias

Vila Real, Março 2008

Orientação de Prof. Doutor João Manuel Pereira Barroso e

Prof. Doutor João Eduardo Quintela Alves de SousaVarajão

Dissertação submetida à Universidade de Trás-os-Montes e Alto Douro como requisito parcial para a obtenção do grau de Mestre em Tecnologias das Engenharias.

A G R A D E C I M E N T O S

Um trabalho de mestrado tem o reconhecido significado a nível académico, mas este representa, acima de tudo, o culminar de um enorme esforço e dedicação, sobretudo porque foi necessária muita persistência para o conseguir conciliar com a minha vida profissional.

Em primeiro lugar, quero agradecer ao meu orientador, Professor Doutor João Varajão, pelo incentivo, pela dedicação, pela excelência da sua orientação e sobretudo pela disponibilidade que sempre demonstrou mesmo estando eu a 400KM de distância.

Agradeço ao meu orientador Professor Doutor João Barroso, pela confiança depositada em mim, por me ter ajudado a arrancar com este projecto e a garantir que ele seria sustentado até ao final.

Um agradecimento especial ao meu marido Luís Saraiva, por ser a minha fonte de inspiração, pelo seu discernimento e sugestão do tema desta tese.

Quero também agradecer ao Centro de Informática da Universidade de Trás-os-Montes e Alto Douro e a todos os colegas da equipa de desenvolvimento, pelo apoio e companheirismo. Não posso deixar de mencionar o meu ilustre colega e amigo Jorge Borges, pelo prazer que tive em trabalhar com ele e pelo muito que me ensinou no início da minha vida profissional.

Finalmente, aos meus pais, aos meus irmãos e restante família, por todo o esforço e investimento que fizeram na minha vida, pela confiança e exemplo.

Espero que este trabalho proporcione tanto prazer a quem o ler, como me deu a mim fazê-lo. Foi um trabalho árduo, mas julgo que os resultados obtidos compensam o esforço despendido.

R E S U M O

A necessidade de integrar Sistemas de Informação sempre foi uma parte fundamental do seu desenvolvimento. No entanto, é indiscutível que nos últimos anos, com a popularização das tecnologias baseadas na *Internet*, a sua importância recebeu um grande impulso.

A integração entre diferentes sistemas informáticos não é um problema novo. Arquitecturas baseadas em mensagens e sistemas de RPC, bem como várias outras tecnologias, das quais se destacam o CORBA, o DCOM e o JavaRMI, foram criadas para tentar resolver esta problemática, mas só com os Web Services é que na realidade a integração se torna efectivamente universal.

Os Web Services apresentam um enorme potencial dado que pela primeira vez se produziu um consenso em torno das normas *standard* a utilizar. A interoperabilidade dos Web Services é sobretudo o reflexo do esforço reunido dos grandes produtores de software e as das organizações que definem as especificações como o WS-I, em torno de um conjunto de linguagens e tecnologias, designadamente: XML, SOAP, UDDI e WSDL.

Perante o seu potencial e após um período de desenvolvimento e maturação, é expectável que os WS venham a emergir como uma das primeiras soluções de interoperabilidade e conectividade mais bem sucedidas, permitindo a redução de custos e a rapidez de desenvolvimento, implementação e integração de todo o tipo de sistemas.

Tendo por referência as características estudadas das arquitecturas orientadas aos serviços (SOA), esta tese propõe um modelo de SI usando Web Services, concretizado através do desenvolvimento de um protótipo que implementa a integração entre o portal TRAS-OS-MONTES.NET e o sistema WSAutarquias.

A B S T R A C T

The need to integrate Information Systems has always been a fundamental part in its development. However, it is undeniable that in recent years with the popularization of technologies based on the *Internet*, its importance has received a great boost.

The integration between different systems is not a new problem. Architectures based on messages and RPC systems, as well as various technologies, which are highlighted the CORBA, DCOM and JavaRMI, were created to try to solve this problem, but only with the Web services is that in fact the integration becomes truly universal.

The Web services have enormous potential because for the first time there produced a consensus around the standard protocols to be used. Interoperability of Web Services is primarily a reflection of the effort meeting of the major producers of software and the organizations that define the specifications such as WS-I, around a set of languages and technologies including: XML, SOAP, WSDL and UDDI.

Given its potential and after a period of development and maturation, it is expected that the WS will emerge as one of the first solutions for interoperability and connectivity more successful, allowing the reduction of cost and speed of development, implementation and integration of the whole type of systems.

Having by reference the studied characteristics of the service oriented architectures (SOA), this thesis proposes a model of SI using Web Services, implemented through the development of a prototype that implements the integration of the portal TRAS-OS-MONTES.NET and the WSAutarquias system.

Í N D I C E

AGRADECIMENTOS I
RESUMO I
ABSTRACT I
ÍNDICE I
ÍNDICE DE FIGURAS II
SIGLAS II

1. Introdução 1

1.1 Enquadramento 1
1.2 Motivações, objectivos e método 5
1.3 Organização da dissertação 6

2. Aplicações distribuídas 9

2.1 Motivações para o desenvolvimento de aplicações distribuídas 9
2.2 Características de aplicações distribuídas 11
2.3 Integração e interoperabilidade em aplicações distribuídas 15
2.4 Arquitecturas 19
2.5 Tecnologias de aplicações distribuídas 27
2.6 Resumo 30

3. Web Services 33

3.1 Definição de Web Service 33
3.2 XML 35
3.3 SOAP 39
3.4 UDDI 44
3.5 WSDL 46
3.6 Interoperabilidade 49
3.7 Resumo 52

4. Proposta de um Sistema de Informação usando Web Services 53

4.1 Modelo conceptual 53
4.2 Modelo de arquitectura 55
4.3 Modelo tecnológico 62
4.4 Apresentação do projecto 65
4.5 Protótipo de integração 67
4.6 Verificação do protótipo 77
4.7 Resumo 82

5. Conclusões 83

REFERÊNCIAS 89

Anexo I - Descrição WSDL dos Web Services do sistema WSAutarquias 95
Anexo II - Código que implementa a integração, o sistema de segurança no cliente e a utilização de SSL 165
Anexo III - Código que implementa a integração de modo síncrono e assíncrono 171

Í N D I C E D E F I G U R A S

Figura 1: O cenário do projecto antes da integração 4

Figura 2: O cenário do projecto após a integração 5

Figura 3: Documento XML "bem-formado" 36

Figura 4: Um possível DTD para o XML da figura 3 37

Figura 5: Um possível XML *Schema* para o XML da figura 3 38

Figura 6: Representação de uma mensagem SOAP 42

Figura 7: Exemplo de pedido SOAP sobre HTTP 43

Figura 8: Exemplo de resposta SOAP sobre HTTP 43

Figura 9: Representação de um documento WSDL 47

Figura 10: Estrutura genérica de um documento WSDL 49

Figura 11: WS-I, *standards* e indústria 51

Figura 12: Arquitectura orientada aos serviços 56

Figura 13: Arquitectura baseada em Web Services 58

Figura 14: Modelo de seis camadas de uma arquitectura usando Web Services 61

Figura 15: Suporte tecnológico de uma arquitectura baseada em Web Services 62

Figura 16: Visão geral do protótipo de integração 66

Figura 17: Modelo de entidades do protótipo de integração 68

Figura 18: Arquitectura do protótipo de integração 71

Figura 19: Web Methods de WSAutarquias 73

Figura 20 : Directiva ASPX para code behind 74

Figura 21: Diagrama de sequência de segurança 75

Figura 22: Lista das SOAP *Exceptions* de WSAutarquias 76

Figura 23: Página inicial no portal TRAS-OS-MONTES.NET 78

Figura 24: Página de entidades aderentes no portal TRAS-OS-MONTES.NET 79

Figura 25: Página de serviços de uma entidade no portal TRAS-OS-MONTES.NET . 80

Figura 26: Página de processos no portal TRAS-OS-MONTES.NET 80

Figura 27: Página de visualização de processo no portal TRAS-OS-MONTES.NET .. 81

Figura 28: Fases da adopção dos Web Servives 85

S I G L A S

CORBA	Object Management Group
DCOM	Distributed Component Object Model
DTD	Document Type Declaration
EAI	Enterprise Application Integration
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDL	Interface Definition Language
IETF	Internet Engineering Task Force
JVM	Java Virtual Machine
MIME	Multipurpose Internet Mail Extensions
OASIS	Organization for the Advancement of Structured Information Standards
ORB	Object Request Broker
RPC	Remote Procedure Call
SGML	Standard Generalized Markup Language
SI	Sistemas de Informação
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol / Internet Protocol
UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WS	Web Service
WSDL	Web Services Definition Language

WS-I	Web Services Interoperability Organization
WWW	World Wide Web
XML	Extensible Markup Language
XSD	XML Schema Definition
XSLT	XML Transformations

C A P Í T U L O I

1. Introdução

Neste primeiro capítulo contextualiza-se o tema da dissertação e apresenta-se o problema a resolver que se insere na área dos Sistemas de Informação distribuídos baseados na *Internet*. Apresenta-se também o cenário da solução de integração proposta, no portal que serve de suporte a esta tese, denominado TRAS-OS-MONTES.NET.

Serão focados, em particular, aspectos ligados à interoperabilidade entre sistemas heterogéneos, bem como aspectos genéricos sobre um dos principais desafios para as Tecnologias de Informação, que tem sido, e continuará a ser, a integração de Sistemas de Informação (SI).

De seguida, apontam-se as motivações, objectivos e a abordagem seguida na elaboração desta dissertação, bem como se faz uma breve descrição da organização da dissertação, referindo-se sucintamente o conteúdo de cada um dos capítulos.

1.1 Enquadramento

Ao longo do tempo, as organizações têm implementado aplicações com complexidade crescente, por forma a responder à evolução dos processos de negócio.

Não obstante a necessidade de comunicação entre departamentos e organizações, a generalidade das aplicações foram sendo disponibilizadas isoladamente, com lógicas, linguagens e bases de dados específicas e incapazes de se compreenderem de parte a parte.

Neste momento assiste-se a uma nova evolução determinada pelo descentralizar do processamento e pelo armazenamento de dados em localizações físicas diferentes, não sendo a função lógica determinada pela topologia física. Contudo, mesmo não

estando o processamento e o armazenamento de dados centralizado, a lógica continua a poder estar numa única aplicação, não obstante estar desenhada de forma distribuída.

É neste contexto que a capacidade de integrar Sistemas de Informação, isto é, de construir sistemas complexos a partir da composição integrada de subsistemas e componentes diversos, se revela fundamental para o futuro.

À medida que as organizações se tornam menos centralizadas e se consolida o envolvimento com parceiros, clientes e fornecedores, aumenta a procura de tecnologias versáteis de interligação.

As soluções disponíveis são ainda reduzidas, mas a disseminação de formatos que promovem a integração, como o XML, de conceitos como o *Enterprise Application Integration* (EAI), têm sido fulcrais para a integração de múltiplas aplicações e bases de dados [IDC 2002].

O final dos anos 90 marca o ponto de partida para a rápida adopção de *standards Web* como WWW, HTTP e XML, o que nos leva a repensar a forma como os nossos Sistemas de Informação são desenvolvidos. Cada vez mais, se torna necessário que as novas aplicações sejam pensadas de forma integrada e distribuída, permitindo a interoperação de sistemas heterogéneos [Silva 2003].

No contexto desta necessidade de integração encontra-se o projecto TRAS-OS-MONTES.NET, um sub-projecto do projecto Trás-os-Montes Digital, originalmente designado por Serviço Cooperativo de Extensão em Trás-os-Montes e Alto Douro (SCETAD) [Bulas Cruz e Ramos 2001]. Este insere-se no quadro das iniciativas do Governo orientadas para a Sociedade de Informação, nomeadamente no âmbito do programa "Cidades e Regiões Digitais" [POSI 2003].

No âmbito das Cidades Digitais, apresenta-se um excerto do discurso do Senhor Ministro da Ciência e Tecnologia, a 10 de Fevereiro de 1998 em Aveiro, o qual revela a elevada necessidade de interoperabilidade entre sistemas de informação [Espigueiro 2002]:

> *"A aplicação de forma integrada de serviços digitais pode, se conduzida com o objectivo da melhoria da qualidade de vida de todos os cidadãos, ser decisiva para o futuro de muitas cidades. A utilização de tecnologias digitais de informação e de telecomunicação para a melhoria dos cuidados de saúde, a efectiva redução da burocracia administrativa, a capacidade de geração de trabalho qualificado e de tele-trabalho, a simplificação e transparência dos processos de decisão, a qualidade e diversidade da informação recebida ou tratada, a abertura e reconhecimento dos processos de educação e de formação profissional, a generalização segura do comércio electrónico, a oferta de novos modos de lazer, o apoio a cidadãos com necessidades especiais, entre muitas outras dimensões, são elementos constitutivos do modelo da "Cidade Digital"".*

A finalidade do projecto TRAS-OS-MONTES.NET, enquanto portal de serviços, consiste em oferecer ao cidadão uma interface Web comum, para um amplo conjunto de serviços de várias entidades, pretendendo deste modo contribuir para uma melhoria do acesso à informação, particularmente aos serviços que cada entidade dispõe [UTAD 2004].

Este projecto concretizou-se, numa primeira fase, tendo como entidades prestadoras de serviços Câmaras Municipais, Sub-Regiões de Saúde, a Direcção Regional de Agricultura e Centros Distritais da Solidariedade e Segurança Social, sendo que cada uma das entidades disponibilizou entre dois a trinta e sete dos seus serviços on-line.

Os serviços colocados à disposição no portal TRAS-OS-MONTES.NET propiciam a troca de informações entre munícipe e os principais serviços descentralizados da Administração Pública. Utilizando formulários específicos, os cidadãos podem submeter pedidos e acompanhar o seu workflow administrativo.

O Centro de informática da Universidade de Trás-os-Montes e Alto Douro (Ci-UTAD) tem a seu cargo a tarefa de desenvolvimento e manutenção da plataforma tecnológica do projecto, bem como do acompanhamento de todos os processos que lhe são inerentes, funcionando como elemento de ligação entre as várias entidades parceiras.

O Sistema de Informação no qual assenta o portal TRAS-OS-MONTES.NET, desenvolvido e mantido por esta equipa, possui a sua própria base de dados, componentes de *middleware*, aplicações de manutenção e aplicações de monitorização entre outras, disponibilizadas através de um *Data Center* que se encontra na UTAD.

Por outro lado, as entidades parceiras dispõem dos seus próprios Sistemas de Informação, aplicações vocacionadas para utilização interna, sem qualquer ligação a recursos externos à entidade, quer ao nível de equipamentos, quer ao nível dos dados.

Apresenta-se de seguida, na figura 1, um diagrama com o cenário existente. Note-se a falta de ligação entre os sistemas informáticos das diferentes instituições. Esta solução utiliza somente o servidor local como repositório de todas as informações e não se integra com os SI das entidades prestadoras de serviços.

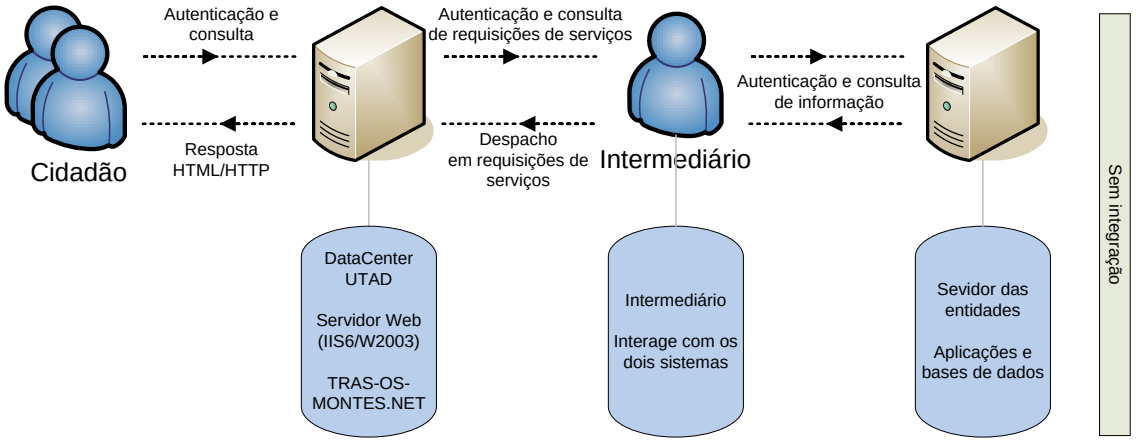

Figura 1: O cenário do projecto antes da integração

Na figura anterior, o utilizador "intermediário" representa um técnico de uma entidade, este acede ao portal TRAS-OS-MONTES.NET para consultar os pedidos efectuados à instituição. O intermediário interage com os dois sistemas de modo a criar o pedido no SI da instituição que representa e actualizar o estado desse pedido no portal TRAS-OS-MONTES.NET.

Desta forma, foi identificado um problema de troca de informações entre os Sistemas de Informação destas instituições e a UTAD. Contudo, verificou-se que, no conjunto de entidades parceiras, existe um grande número de casos que utiliza produtos

de *software* provenientes do mesmo fornecedor, revelando assim uma oportunidade de desenvolvimento de um sistema de integração comum.

A solução de integração deverá permitir a interacção entre o portal de serviços orientado aos cidadãos, TRAS-OS-MONTES.NET, e os Sistemas de Informação específicos das Administrações Municipais onde se encontram os processos dos munícipes que se pretendem consultar em tempo real.

De seguida, na figura 2, apresenta-se a ligação que é desejada e necessária criar, elucidando alguns aspectos relativos à abordagem a adoptar. Deste modo, oferece-se ao cidadão a possibilidade de consultar os seus processos de forma organizada, sem ter de se dirigir fisicamente a cada uma das instituições, em busca dessas informações.

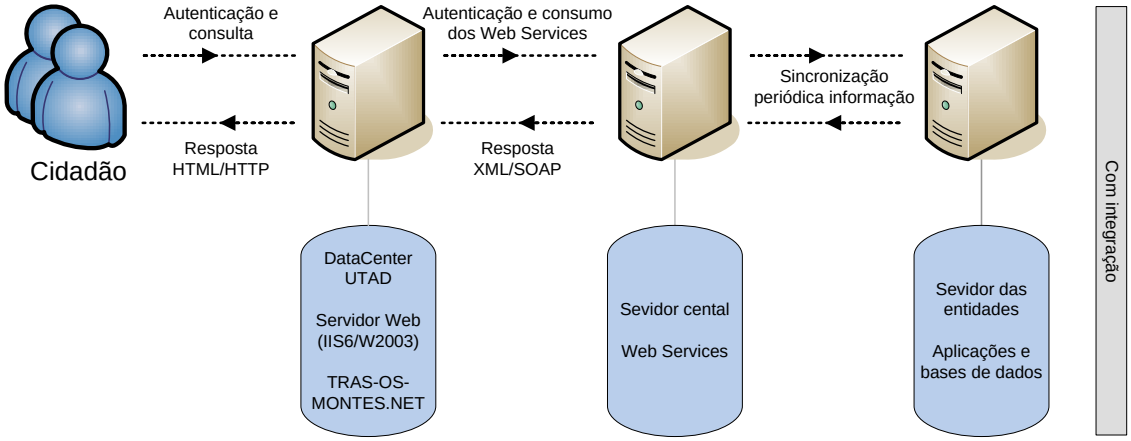

Figura 2: O cenário do projecto após a integração

A implementação da solução de integração justifica-se por uma série de vantagens obtidas quer por parte das entidades, quer por parte do cidadão.

1.2 Motivações, objectivos e método

As motivações para o desenvolvimento do sistema aqui apresentado advêm da necessidade de se criar uma plataforma de integração entre diversos Sistemas de Informação que não se limite simplesmente à troca de dados, sendo fundamental encontrar uma forma de utilizar serviços disponíveis em plataformas heterogéneas.

Conjugada com esta necessidade, houve também a motivação de aprofundar os conhecimentos em termos das questões envolventes aos conceitos de interoperabilidade, ao mesmo tempo que se pretendia investigar a utilização de um novo paradigma tecnológico, os Web Services.

Capacitar Sistemas de Informação diferentes para trocar informação entre si de forma *standard*, independentemente do sistema operativo, da linguagem de programação e da localização dos dados, é sem dúvida uma questão fascinante e de grande relevância.

O objectivo central deste trabalho é a apresentação de uma arquitectura conceptual e técnica, suportada por um protótipo funcional, que permita a conectividade e interoperabilidade entre Sistemas de Informação baseados na Web. A solução de integração destina-se a ser usada e integrada em plataformas bastante díspares, dado que cada entidade possui um Sistema de Informação próprio, que teria de alguma forma de integrar esta nova capacidade.

Sendo que esta dissertação surge num contexto de agitação ao nível dos Sistemas de Informação distribuídos, é objectivo específico estudar a arquitectura de software orientada a serviços e as tecnologias envolventes estabelecendo as limitações e as capacidades deste novo modelo.

A presente dissertação foi conduzida segundo o método tradicional, em três etapas fundamentais: identificação e formulação do problema a resolver; revisão bibliográfica de modo a diagnosticar as necessidades de aprendizagem, a recolha das referências e o estudo que se propõe; apresentação de uma proposta de SI que resolva o problema identificado e concretização de um protótipo com o modelo proposto; e, finalmente, discusão das conclusões obtidas com o trabalho realizado.

1.3 Organização da dissertação

A dissertação foi estruturada em cinco capítulos distintos: *Introdução*, *Aplicações Distribuídas*, *Web Services*, *Proposta de um SI usando Web Services* e,

finalmente, *Conclusões*. De seguida, faz-se uma breve descrição do conteúdo de cada um deles.

Neste primeiro capítulo procede-se ao enquadramento e caracterização sumária do problema que levou à formulação da tese e ao desenvolvimento da presente dissertação. Apontam-se também as motivações, objectivos e a abordagem seguida na sua realização.

No capítulo 2 (*Aplicações Distribuídas*) descrevem-se as características e os conceitos de integração e interoperabilidade de aplicações distribuídas, bem como os desafios enfrentados na construção destes sistemas. Em seguida, procede-se a uma análise sobre as várias tecnologias actualmente disponíveis para integração de SI, nomeadamente para a construção de soluções baseadas em aplicações distribuídas.

No capítulo 3 (*Web Services*) aborda-se a definição e fundamentos teóricos dos *Web Services*, apresentando-se detalhadamente as principais tecnologias associadas, nomeadamente XML, SOAP, UDDI e WSDL. Referem-se também as principais organizações que se empenham em definir e publicar as especificações destas tecnologias.

No capítulo 4 (*Proposta de um SI usando Web Services*) apresenta-se uma plataforma de integração que permite a conectividade e interoperabilidade entre aplicações distintas. Descreve-se a arquitectura do sistema desenvolvido e mostra-se como a tecnologia Web Services se ajusta perfeitamente na arquitectura proposta.

Finalmente, no capítulo 5 (*Conclusões*), salientam-se as conclusões mais importantes, as contribuições deste trabalho e as perspectivas para o futuro. Identificam-se os principais desafios que se colocam às soluções de integração baseadas em Web Services e as fases de adopção dos Web Services ao longo do tempo.

C A P Í T U L O I I

2. Aplicações distribuídas

A crescente popularidade da *Internet* e o crescimento do número de serviços e informações disponibilizadas nesse meio, leva-nos a repensar a forma como os nossos Sistemas de Informação são desenvolvidos. Cada vez mais, torna-se necessário que as novas aplicações sejam concebidas de forma integrada e distribuída, permitindo às mesmas interoperar com sistemas distintos e heterogéneos.

Neste capítulo começa-se com uma breve apreciação das diversas motivações que levam ao desenvolvimento de aplicações distribuídas. Para, de seguida, se descreverem os principais conceitos e características das mesmas. Depois deste enquadramento passa-se à apresentação das várias tecnologias actualmente disponíveis para soluções baseadas em aplicações distribuídas, bem como alguns cenários de utilização.

2.1 Motivações para o desenvolvimento de aplicações distribuídas

Até aos últimos anos a maioria dos grandes SI eram centralizados, caracterizados pela existência de um computador servidor (*mainframe*) que funcionava normalmente em modo *batch* com terminais conectados a ele. Esses terminais tinham pouca ou nenhuma capacidade de processamento, de modo que todo o processamento de informações era da responsabilidade do computador *mainframe* [Sommerville 2001].

Actualmente assistimos a uma transformação fundamental do mercado informático e inúmeras motivações apontam para a descentralização das aplicações, entre elas incluem-se as apresentadas nos pontos seguintes [Tanenbaum 2002].

- O elevado custo dos *mainframes*

  Não só porque o elevado investimento inicial já é um risco para as empresas, mas também porque havendo um único ponto, a falha desse pode ser um risco insuportável.

- A posse da informação

  Um importante factor associado à descentralização é a política de posse da informação. Departamentos, divisões, empresas geograficamente distribuídas que contenham dados não querem ceder essa informação para uma outra localização central.

- Segurança

  Para qualquer organização dados confidenciais devem estar seguros, no entanto precisam de ser facilmente acessíveis. Estes dois requisitos são mais facilmente conseguidos se os dados estiverem fisicamente segmentados.

Os factos precedentes alertam para a emergência de um novo padrão de desenvolvimento de aplicações, conhecido como sistema distribuído.

Um sistema distribuído é constituído por uma colecção de computadores distintos, capazes de distribuir o processamento da informação entre si e vistos pelos utilizadores como um só [Tanenbaum 2002].

Efectivamente torna-se desejável descentralizar quer o processamento quer o armazenamento de dados, todavia, a lógica continua a poder estar numa única aplicação, desenhando-a como uma aplicação distribuída, cujos requisitos de processamento devem ser satisfeitos por múltiplos computadores e onde a informação possa residir em distintas localizações, mas onde as funcionalidades lógicas não sejam determinadas pela topologia física que é utilizada para implementar a aplicação.

Hoje em dia os sistemas distribuídos são amplamente utilizados, no entanto apercebemo-nos que a indústria de computadores ainda não tirou total partido da sua utilização. À medida que os sistemas de redes sem fios de alta velocidade se tornam amplamente disponíveis, é possível integrar serviços dinamicamente, considerando os componentes distribuídos de uma aplicação como fornecedores de serviços para uma outra aplicação.

O conceito de distribuir funcionalidades engloba a premissa de reutilização. Os programadores podem utilizar cada uma destas funcionalidades como um bloco para muitas aplicações.

2.2 Características de aplicações distribuídas

Seis importantes características dos sistemas distribuídos são [Coulouris *et al.* 1994]:

- Partilha de recursos

  Um sistema distribuído permite a partilha de recursos de *hardware* e *software*, como discos, impressoras, ficheiros e compiladores, que estão associados a diferentes computadores de uma rede. A partilha de recursos é também possível em sistemas para multi-utilizadores, mas, nesse caso, todos os recursos têm de ser fornecidos e geridos por um computador central.

- Abertura

  A abertura de um sistema diz até que ponto ele pode ser ampliado, adicionando novos recursos não proprietários a ele. Os sistemas distribuídos são sistemas abertos, que normalmente incluem *hardware* e *software* de diferentes fabricantes.

- Concorrência

  Num sistema distribuído, vários processos podem operar ao mesmo tempo em diferentes computadores na rede. Esses processos podem comunicar uns com os outros, durante a sua operação normal.

- Escalabilidade

  Em princípio, pelo menos, os sistemas distribuídos são escaláveis, no sentido de que a capacidade do sistema pode ser aumentada pelo acréscimo de novos recursos. Na prática, a escalabilidade pode ser limitada pela rede, se muitos computadores forem acrescentados, então a capacidade da rede poderá tornar-se inadequada.

- Tolerância a defeitos

  A disponibilidade de vários computadores e o potencial de duplicar informações significa que os sistemas distribuídos podem ser tolerantes a algumas falhas de *hardware* e *software*. Na maioria dos sistemas distribuídos, um serviço de má qualidade pode resultar da ocorrência de falhas; a perda completa de serviço tende a ocorrer somente quando existe uma falha de rede.

- Transparência

  A transparência significa que o utilizador não necessita de conhecer a natureza distribuída do sistema para ter acesso aos seus recursos.

Naturalmente, os sistemas distribuídos apresentam algumas desvantagens [Coulouris *et al.* 1994]:

- Complexidade

  Os sistemas distribuídos são mais complexos do que os sistemas centralizados, dificultando a compreensão das suas propriedades e a realização de testes desses sistemas. Por exemplo, em vez do desempenho do sistema depender da velocidade de um processador, ele depende da largura de banda da rede e da velocidade dos diferentes processadores. Mover recursos de uma parte do sistema para outra pode afectar radicalmente o desempenho do sistema.

- Protecção

  Torna-se mais difícil gerir a protecção num sistema distribuído, pelo facto de poder ser acedido a partir de vários computadores diferentes e o tráfego na rede estar sujeito à intercepção.

- Facilidade de gestão

  Os diferentes computadores num sistema distribuído podem ser de diferentes tipos e podem executar diferentes versões do sistema operativo. Os defeitos numa máquina podem-se propagar para outras máquinas resultando em consequências inesperadas. Isso tudo significa a necessidade de maior esforço para gerir e manter o sistema em operação.

- Imprevisibilidade

  Os sistemas distribuídos são imprevisíveis nas suas respostas, essas respostas dependem da carga total sobre o sistema, da sua organização e da carga na rede. Como tudo isso pode mudar, em curto tempo, o tempo de resposta à solicitação de um utilizador pode variar repentinamente de uma solicitação para outra.

Além das principais vantagens e desvantagens dos sistemas distribuídos, identificam-se de seguida os principais aspectos a ter em consideração no seu desenho [Coulouris *et al.* 1994].

- Identificação de recursos

  Os recursos de um sistema distribuído estão dispersos em diferentes computadores e, portanto, as aplicações devem conter mecanismos para a sua localização e uso eficiente. Um esquema de nomes deve ser constituído para que os utilizadores possam descobrir e referir-se aos recursos de que necessitam. Um exemplo desse esquema de denominações é o URL (*Uniform Resource Locator*), que é utilizado para identificar páginas Web. Se um esquema de identificação significativo e universalmente conhecido não for utilizado, muitos desses recursos serão inacessíveis aos utilizadores de SI.

- Comunicações

  A disponibilidade universal da *Internet* e a eficiente implementação dos protocolos de comunicação TCP/IP (*Transmission Control Protocol / Internet Protocol*), propõem que, para a maioria dos sistemas distribuídos, este é o modo mais eficaz dos computadores comunicarem entre si. Contudo, quando houver requisitos específicos de desempenho, fiabilidade, entre outros, poderão ser utilizadas abordagens alternativas para as comunicações.

- Qualidade do serviço

  A qualidade do serviço oferecido por um sistema reflecte o seu desempenho, a sua disponibilidade e a sua fiabilidade. Ela é afectada por uma série de factores, como a alocação de processos em memória no sistema, a distribuição de recursos, a rede, o hardware e adaptabilidade do sistema.

- Consistência de dados

  Num sistema distribuído, geralmente existem múltiplas cópias dos dados espalhados pela rede, por razões de fiabilidade e melhoria no desempenho (nos acessos aos dados). Assim, mecanismos de partilha e bloqueio devem estar presentes nas aplicações.

- Arquitecturas de *software*

  A arquitectura de *software* descreve como a funcionalidade da aplicação é distribuída numa série de componentes lógicos e como esses componentes são distribuídos nos processadores. Escolher a arquitectura certa para uma aplicação é essencial para atingir a qualidade de serviço desejada.

No desenho de aplicações distribuídas alguns problemas poderão surgir, sendo que esses problemas não são específicos de nenhuma arquitectura em particular [Microsoft 2001a].

- Diferentes tipos de dados

  Por vezes, não existe cem por cento de compatibilidade entre tipos de dados (*data types*) em sistemas operativos diferentes, de modo que, deve ser considerada uma forma de manipular os tipos de dados incompatíveis.

- Falha no servidor

  Como os componentes de um sistema distribuído podem depender de vários servidores existem mais pontos de falha, além disso, a falha de um único ponto pode causar a falha de todo o sistema. Por tudo isto, deve ser considerada uma forma de manipulação dessas falhas e de perda de resposta por parte dos servidores.

- Falha no cliente

  Se um servidor estiver a armazenar o estado de um cliente e o cliente falha, então o servidor deve ser notificado. Por outro lado, deve também considerar-se a necessidade de libertar recursos no servidor que estavam a ser utilizados no cliente.

- Várias chamadas

  Se um método é chamado mas não obtemos resposta por parte do servidor, por vezes, pode não ser aceitável voltar a chamar esse método. Por exemplo, se um método é chamado para reservar uma grande quantidade de *stock* e se o servidor recebe o pedido mas a resposta a esse pedido é perdida, então não será aceitável voltar a submeter esse pedido.

- Segurança

  Em aplicações distribuídas existem mais oportunidades de ataque à segurança. Não só se deve considerar a autenticação e autorização, mas também, como tornar segura a comunicação entre o cliente e o servidor, e como salvaguardar o sistema de eventuais ataques.

- Sincronismo de relógio

  Os diversos relógios utilizados no sistema têm de ser mantidos em sincronia e as aplicações têm de estar preparadas para os problemas de temporização, devido a atrasos ocasionados por congestionamento na rede ou falha de um servidor. Muitas operações são realizadas com base na data e na hora, portanto, deve assegurar-se o sincronismo do relógio, por exemplo, não é aceitável para um servidor que uma ordem de compra tenha sido concretizada antes da mesma ter sido solicitada.

2.3 Integração e interoperabilidade em aplicações distribuídas

A necessidade da integração tem sido um tema permanentemente em agenda na área de Sistemas de Informação, especialmente desde a década de 80. Uma aplicação

tem tanta mais utilidade quanto mais e melhor integração tiver com outras aplicações, além disso todos os SI deverão ter todas as suas aplicações integradas [Silva 2003], sempre que tal se justificar.

Por integração entende-se a complementaridade entre componentes e uma fácil interoperação entre eles. Quer isto dizer que, a integração visa a criação de condições para que os vários componentes, independentemente do seu nível de autonomia, possam dialogar e cooperar com vista a atingir os objectivos do sistema.

Considerando um SI como um conjunto de entidades físicas e lógicas organizadas para a realização de determinados objectivos [Matos 2001], o processo de integração visa contribuir para uma optimização da cooperação entre essas entidades.

A integração de sistemas pode ser encarada sob diferentes perspectivas consoante os aspectos a tratar. Alguns dos mais significativos são expostos de seguida [Matos 2001].

- Nível funcional

  Em que se visa o estabelecimento dum conjunto coerente de funções ou actividades, complementares entre si, e o encadeamento da sua aplicação, sob a forma de processos, para a realização de cada objectivo.

- Nível da informação

  Sendo que cada componente ou subsistema produz e consome informação, a harmonização de modelos (construção de abstracções comuns) e a eliminação de redundâncias e inconsistências são alguns dos passos na integração de informação.

- Nível de interface e comunicação

  Nesta perspectiva focam-se as necessidades de comunicação para a interoperação entre componentes e subsistemas. A definição de protocolos e mecanismos comuns de interface é uma das etapas na integração de sistemas. As formas de acoplamento entre componentes (forte, fraco, distribuído), os canais físicos de suporte à comunicação (redes) bem como a questão da interface entre o utilizador e a máquina fazem também parte desta perspectiva.

- Nível da coordenação

  O efectivo aproveitamento dum conjunto integrado de entidades dependerá do adequado planeamento da participação de cada uma delas nos processos em curso no sistema. O estabelecimento duma estrutura de coordenação global é pois um requisito da integração de sistemas.

Outra das inúmeras formas de classificar a integração de Sistemas de Informação, e talvez a mais simples, é baseada na geografia [Silva 2003]:

- Integração no computador

  Quando os SI partilham a mesma máquina, seja um *mainframe*, servidor departamental, computador pessoal, dispositivo móvel, máquina de análises, balança, telemóvel ou relógio.

- Integração na empresa

  Quando os SI residem em máquinas distintas (e portanto não podem partilhar a mesma memória) mas estão ligados através de uma rede de banda larga, segura e fiável, de pequena latência.

- Integração entre empresas

  Quando os SI estão ligados ocasionalmente, quando existem restrições de segurança ou quando são geridos por entidades distintas.

Note-se que, tal como em todas as classificações, não existem fronteiras claras e bem definidas entre os três tipos de integração. Por um lado, as condições típicas da integração na empresa também podem existir entre empresas ligadas por circuitos dedicados. Por outro lado, também pode acontecer que dentro de uma empresa existam as condições típicas da integração entre empresas. É o caso, por exemplo, de qualquer empresa com presença física dispersa ou departamentos autónomos, como na Administração Pública. Esta classificação deve, por isso, ser entendida em termos tecnológicos e não administrativos.

A integração enquanto facilitadora da interoperação entre sistemas e da sua coordenação global representa um importante pré-requisito para garantir flexibilidade e

agilidade, ou seja, capacidade de suporte a diferentes processos e a adaptação rápida a alterações imprevistas, quer do ambiente externo, quer interno.

Por outro lado, ao evitar actividades desnecessárias, nomeadamente a repetição da introdução de informação que deve transitar dum subsistema para outro, e também ao permitir uma melhor chegada da informação a todas as áreas das empresas e organizações, a integração constitui uma base para uma maior co-responsabilização e participação de todos os actores intervenientes.

A automatização dos processos de troca de informação, incluindo os necessários mapeamentos, contribui para o aumento da qualidade ao reduzir potenciais erros causados pela introdução manual de informação. A maior fluidez (em termos de informação e sequências de controlo) também contribui para um aumento da qualidade em termos de tempos de resposta do sistema e para que as decisões possam ser tomadas com base em informação actualizada.

Além destas importantes razões para a integração outros benefícios indirectos podem ser citados como uma maior robustez face à aquisição, à estruturação, à gestão e partilha de conhecimento dentro das organizações (*corporate knowledge management*).

A integração de sistemas constitui, em geral, um processo complexo, sujeito a um grande número de obstáculos, de onde se destaca a crescente heterogeneidade de ambientes. Os factores que contribuem para um elevado nível de heterogeneidade são essencialmente as tecnologias de suporte e implementação, como a diversidade de protocolos de rede, arquitecturas de hardware, sistemas operativos ou linguagens de programação.

À medida que as organizações mudam de um modelo baseado em *mainframes* para ambientes abertos e de sistemas distribuídos, questões de interoperabilidade têm recebido crescente atenção. No campo das aplicações o tema interoperabilidade tem-se tornado cada vez mais relevante [Manola 1995].

A procura por interoperabilidade é proporcional à construção de sistemas distribuídos e à medida que os sistemas de *software* e *hardware* se tornam mais

complexos, a interoperabilidade entre diferentes componentes é um factor crítico [Seetharaman 1998].

Wegner [Wegner 1996] define interoperabilidade como sendo a habilidade de dois ou mais componentes de *software* cooperarem, independentemente das suas diferenças em linguagem, interface e plataforma de execução. Heiler [Heiler 1995] define o termo como a habilidade de sistemas distribuídos trocarem serviços e dados entre si.

Embora as duas definições sejam muito semelhantes têm abordagens diferentes. A primeira evidencia componentes de *software* de natureza diferente e provavelmente sobre plataformas distintas. A segunda enfatiza a funcionalidade (troca de serviços e informações) sem se preocupar com os detalhes de heterogeneidade destes sistemas. São perspectivas que não se contradizem mas que, pelo contrário, se complementam.

Existem outras definições para o termo, mas todas concordam que a interoperabilidade não se consegue naturalmente, e por essa razão deve basear-se em padrões aceites o mais amplamente possível pelo mercado, em vários itens, como protocolos, serviços e requisições. O facto é que se a interoperabilidade é um conceito simples, possui ao mesmo tempo uma complexidade proporcional à abrangência em que é aplicada.

2.4 Arquitecturas

O desenvolvimento de aplicações distribuídas requer novas técnicas de desenho e novos modelos. Nesta secção abordam-se os dois modelos de arquitecturas existentes para efectuar a troca de informações, mantendo a interoperabilidade entre tecnologias e sistemas (*middlewares*), são ainda discutidas as características destas arquitecturas e finalmente, mostra-se o efeito dos *standards* Web no desenvolvimento de aplicações distribuídas.

2.4.1 Arquitectura baseada em mensagens

A arquitectura baseada em mensagens (*message-based architectures*), apresenta-se como uma séria candidata para o desenvolvimento de aplicações distribuídas. Esta tecnologia utiliza, na maior parte dos casos, o conceito de filas de espera (*waiting queues*) para trocar dados, implementa-se recorrendo a um único canal de comunicação ao qual se conectam todas as aplicações [Microsoft 2001a].

O canal de comunicação deve ser independente de qualquer tipo de hardware, linguagens de programação, sistema operativo e tecnologia de rede para estar acessível a todas as aplicações.

A grande vantagem desta tecnologia é que suporta mensagens assíncronas. Isto é, o canal armazena a mensagem numa fila de espera quando a aplicação de destino não está disponível para receber essa mensagem, deste modo é permitido à aplicação que envia a mensagem continuar a execução do seu trabalho logo após o seu envio, não ficando parada à espera da entrega ou de qualquer tipo de comunicação da rede.

Estas filas são de grande utilidade, pois ao contrário das tecnologias síncronas, a aplicação destino não necessita de estar sempre disponível para receber as mensagens. É possível que a aplicação destino receba e processe as mensagens quando ela própria o desejar, como, por exemplo, em alturas de menos trabalho, ou atendendo a uma lista de prioridades estabelecida.

Cada aplicação remetente tem à sua responsabilidade a construção da funcionalidade adicional de empacotar, desempacotar, classificar e validar o conteúdo da mensagem. À medida que a complexidade e flexibilidade das mensagens aumenta, desempacotar e validar mensagens torna-se difícil e tende a ser dispendioso.

A maior parte dos SI baseados em mensagens são implementados utilizando produtos de *Message Queuing* proprietários, o formato (ou estrutura) da mensagem, assim como a sua interpretação não estão normalizados. Esta lacuna tem como consequência que, uma vez escolhido determinado produto, é muito difícil adoptar outro produto de outro fabricante, além disso os diversos produtos no mercado não são à

partida integráveis, criando enormes dificuldades, por exemplo, quando se fundem duas empresas.

Por outro lado, em muitos cenários de aplicações distribuídas são invocados *workflows* que definem uma determinada sequência de troca de mensagens. Todavia, como as mensagens são enviadas de forma assíncrona, é possível que cheguem fora da ordem, provocando graves problemas quando as mensagens são processadas numa sequência incorrecta. Para resolver este problema é necessário criar uma aplicação adicional ou um protocolo de alto nível em cima do protocolo de *messaging* para apanhar a sequência correcta das mensagens.

Recentemente, por razões de *marketing* os novos produtos baseados em mensagens preferem a nova designação "message brokers", embora a base tecnológica seja exactamente a mesma da integração com mensagens.

2.4.2 Arquitectura baseada em RPC

A integração entre Sistemas de Informação baseada em chamadas a procedimentos remotos, RPC, de *Remote Procedure Call*, apareceu no início dos anos 80 como consequência quase natural das primeiras linguagens de programação baseadas em procedimentos, como o Pascal e o C.

O seu grande sucesso inicial foi renovado com o aparecimento de novos modelos de programação que utilizam esta arquitectura, onde prevalecem os objectos distribuídos, os componentes de acesso remoto[1] e, mais recentemente, os Web Services. De facto as novas tecnologias são quase todas baseadas em RPC [Silva 2003].

Uma *Remote Procedure Call* é uma chamada ordinária a um procedimento ou função que reside no mesmo ou num distinto SI. Essa comunicação é extremamente simples tanto em termos conceptuais como em termos práticos, pois para o cliente o pedido é feito como se o procedimento remoto fosse local, além disso uma RPC

[1] O termo remoto é normalmente empregue caso a chamada seja direccionada para outro servidor.

providencia localização transparente, ou seja, quem está a chamar o procedimento não necessita de conhecer a localização física do *Service Provider*.

Um procedimento é um conjunto de código que implementa um serviço bem definido e oferece uma interface para que outros procedimentos possam pedir a execução desse serviço. Os procedimentos permitem estruturar melhor um programa e assim facilitam imenso a manutenção e reutilização do código existente.

A infra-estrutura de RPC gera um *stub* (canal de comunicação) também conhecido por *proxy*, que actua como uma representação do código do procedimento remoto e transforma qualquer argumento do procedimento para um *buffer*, o qual pode ser transmitido através da rede para um servidor RPC. No servidor de RPC o *stub* descompacta os argumentos no processo servidor e os argumentos são então passados para a função a ser chamada. Qualquer valor a ser retornado é devolvido para quem chamou a função de uma forma similar [Microsoft 2001a].

Antes de qualquer comunicação, é fundamental que os servidores publiquem as interfaces (nomes, parâmetros e resultados) dos procedimentos disponibilizados, caso contrário, os clientes não saberão que servidores e procedimentos estarão disponíveis para serem chamados remotamente. Estas interfaces são escritas numa linguagem própria chamada IDL, *Interface Definition Language*, que é independente da linguagem de programação quer do cliente quer do servidor, deste modo será possível ao cliente chamar procedimentos remotos escritos numa linguagem de programação diferente. A descrição completa da linguagem IDL pode ser obtida no capítulo três da especificação CORBA [OMG 2001].

Contudo, uma desvantagem está relacionada com a evolução das interfaces dos procedimentos e até dos próprios procedimentos, pois não existe nada na arquitectura RPC que permita avisar os clientes que, apesar da interface ser idêntica, o serviço prestado pelo procedimento foi alterado, ou mais grave ainda, que a interface de determinado procedimento foi alterada.

A arquitectura baseada em RPC tem uma série de vantagens, principalmente quando comparada com a arquitectura baseada em mensagens [Silva 2003].

- Simplicidade

  Como o conceito de procedimento era conhecido dos programadores, e além disso a maior parte deles estão acostumados a utilizar de alguma forma algum tipo de chamadas a funções, o conceito de procedimento remoto foi fácil de aceitar, aprender e utilizar, tanto que se tornou num modelo de programação familiar, ao contrário das mensagens.

- Sincronismo

  Ao contrário da arquitectura (assíncrona) baseada em mensagens, a arquitectura baseada em procedimentos remotos é síncrona, a *thread* de execução é bloqueada até que a função retorne e termine, de tal forma que existe sempre uma resposta com o resultado. Deste modo, o programa cliente tem sempre uma confirmação que os parâmetros foram entregues e que o procedimento executou o serviço pedido, assim como o respectivo resultado desse pedido.

- Granularidade

  Permite a um programa comunicar com um procedimento remoto que é apenas uma parte de outro programa, permitindo focar melhor a comunicação e dirigir o pedido directamente para a zona do programa que trata esse pedido.

- Sinergia com o conceito cliente/servidor

  A arquitectura potencia, e é potenciada, pelo conceito cliente/servidor. Com efeito, a arquitectura RPC é baseada no conceito de um programa servidor que fornece procedimentos para executarem pedidos enviados por outros programas, que são por isso os seus clientes. Logo à partida, esta decisão diferencia esta arquitectura das mensagens, onde cada programa pode trocar mensagens com qualquer outro programa. O modelo cliente/servidor será discutido na secção 2.4.3.

No entanto, apesar do sincronismo da comunicação entre cliente e servidor ser uma das grandes vantagens desta arquitectura, é fundamental garantir que os procedimentos vão mesmo estar disponíveis na altura em que são chamados, ao contrário da arquitectura baseada em mensagens. Com efeito, apesar do sincronismo assegurar que essa comunicação ocorreu dispensando assim outras garantias, um

modelo síncrono numa arquitectura distribuída introduz alguns problemas [Microsoft 2001a]:

- Construção de redundância

  Para que uma aplicação se possa conectar a um ponto para requerer um serviço necessita conhecer os seus parâmetros. A forma encontrada para que isso seja possível, é acrescentar informação adicional ao serviço em *hard-code*. Esta não é uma solução óptima porque torna o desenvolvimento redundante e a capacidade de tolerância a falhas bastante difícil de implementar.

- Distribuição de carga e tolerância a falhas

  Não existe uma forma simples para uma aplicação baseada em RPC executar qualquer tipo de distribuição de carga (*load balancing*) dinamicamente. Além disso, também não consegue uma aplicação responder a uma indisponibilidade de um servidor alternando de uma forma dinâmica para outro servidor.

- Estabelecimento de prioridades

  Estabelecer a prioridade dos pedidos é quase impossível, porque todos os pedidos são atendidos por ordem de chegada. Se um servidor em particular se encontra muito sobrecarregado, os clientes prioritários podem ficar sujeitos a atrasos inaceitáveis.

2.4.3 Modelos de sistemas distribuídos

Nesta secção serão analisados dois tipos genéricos de modelos de sistemas (ou aplicações) distribuídos:

- Modelo cliente/servidor;
- Modelo de objectos distribuídos.

O modelo cliente/servidor representa um sistema como um conjunto de serviços fornecidos por servidores e um conjunto de clientes que utilizam esses serviços [Orfali e Harkey 1998]. Os clientes necessitam de conhecer os servidores que estão disponíveis, no entanto, de uma maneira geral, eles não sabem da existência de outros clientes.

O desenho do modelo cliente/servidor deve reflectir a estrutura lógica da aplicação, dividindo o *middleware* em camadas: dados, funcionalidades e interface com o utilizador [Hirschfeld 1996]. Um exemplo bem conhecido desta abordagem é o modelo proposto por Buxton [Buxton 1980], que sugeriu um modelo de três camadas, a camada de apresentação disponibiliza informações aos utilizadores e mantém todas as interacções com eles, a camada de processamento implementa a lógica da aplicação e a camada de gestão de dados disponibiliza as operações de interacção com os dados (mais frequentemente bases de dados).

Em sistemas centralizados estas operações não precisam de estar nitidamente separadas, contudo, quando se está a projectar um sistema distribuído, deve-se fazer uma clara distinção entre elas, pois, desta forma, torna-se possível distribuir cada camada para um computador diferente. Do mesmo modo, é fácil incluir um novo servidor e integrá-lo no sistema ou fazer a actualização dos servidores de modo transparente, sem afectar o restante sistema.

Paralelamente a este modelo de desenvolvimento em termos de computação, aparece o conceito de componentes. A necessidade das interfaces gráficas com o utilizador (*Grafical User Interface* - GUI) resolverem os problemas da reutilização de código e da manutenção de uma certa coerência, foi um dos factores que originou este desenvolvimento. Após esta utilização inicial bem sucedida, estes elementos foram evoluindo de forma a comportarem utilizações mais complexas, não relacionadas apenas com a interface.

Um componente de *software* pode ser definido como [CBDi 1999]:

> "Uma peça identificável de software que descreve ou fornece um conjunto de serviços significativos e que apenas é usada através de interfaces bem definidas."

A primeira utilização em larga escala de componentes tornou-se realidade em 1991 com o aparecimento, pela mão da Microsoft, do ambiente de desenvolvimento *Visual Basic* [Microsoft 2001b]. Este ambiente concebia a possibilidade de se inserirem

objectos (componentes VBX), programados em C/C++, que adicionavam novas funcionalidades às aplicações.

Rapidamente, floresceu em volta deste ambiente um mercado de componentes para o desenvolvimento de aplicações. Os componentes desenvolvidos internamente ou comprados a terceiros tornaram-se uma forma expedita de implementar ou actualizar funcionalidades nas aplicações.

No modelo de objectos distribuídos não há nenhuma distinção entre servidores e clientes. Podemos pensar no sistema como um conjunto de objectos que interagem e cuja localização é irrelevante. Neste modelo um conjunto de objectos, com interfaces bem definidas, chama serviços oferecidos por outros objectos. Para utilizar serviços, os objectos devem explicitamente fazer referência ao nome e à interface de outros objectos.

Uma vez que os objectos não estão fortemente acoplados, a implementação de objectos pode ser modificada, sem afectar outros objectos. No entanto, se uma modificação na interface for requerida para satisfazer as mudanças propostas no sistema, deverá ser avaliado o efeito dessa mudança em todos os utilizadores do objecto alterado.

Os diversos componentes de um sistema distribuído podem ser implementados em diferentes linguagens de programação e executados em tipos de processadores completamente distintos. Também, os modelos de dados, a representação das informações e os protocolos de comunicação podem, todos eles ser diferentes.

Portanto, um sistema distribuído exige um *software* que possa gerir essas diversas partes e garantir que elas comunicam e trocam dados entre si. O termo *middleware* é utilizado para designar esse *software*, pois ele é o intermediário dos diferentes componentes distribuídos do sistema. Os principais padrões existentes para o *middleware* trabalhar com objectos distribuídos serão descritos na secção 2.5.

2.5 Tecnologias de aplicações distribuídas

Os esforços colocados na integração de sistemas nas últimas décadas estão intimamente ligados aos recentes desenvolvimentos nas tecnologias da informação e comunicação. De facto os avanços nestas tecnologias constituem importantes factores facilitadores da integração de sistemas distribuídos.

Nesta secção realizar-se-á uma abordagem teórica referente às principais tecnologias para sistemas distribuídos, de entre as actuais se destacam a CORBA do *Object Management Group* (OMG) [OMG 2004], o DCOM da Microsoft e o JavaRMI da Sun [Sun 1999]. Reservam-se para abordar no capítulo três os conceitos da nova tecnologia, os Web Services.

2.5.1 CORBA

O esforço de normalização de múltiplas empresas reunidas no consórcio *Object Management Group* (OMG) deu origem à arquitectura CORBA, *Common Object Request Broker Architecture*. Esta foi a resposta criada no início dos anos 90 para fazer face à necessidade de interoperabilidade entre aplicações baseadas em objectos distribuídos.

A base desta arquitectura é um barramento de objectos, denominado ORB, *Object Request Broker*, que é o componente de middleware que fornece uma estrutura de comunicações para os objectos interagirem entre si, independentemente de aspectos específicos da plataforma e das técnicas utilizadas para a sua implementação, isto é, um cliente poder invocar de forma transparente um método de um objecto no servidor, o qual pode estar no mesmo computador ou algures na rede, pois o encaminhamento do pedido é da competência do ORB [Roque 2003].

Em CORBA todos os objectos são definidos por uma interface escrita em IDL, esta linguagem é do tipo declarativa e isenta de ambiguidades, de modo que a interface seja independente da linguagem de programação em que os objectos foram desenvolvidos, contudo existem vários standards de IDL criados pela OMG para linguagens como C, C++, Java, COBOL, Smalltalk, Ada, Lisp, Python e IDLscript.

Em teoria a arquitectura CORBA foi projectada para ser uma excelente solução para sistemas cliente/servidor heterogéneos, no entanto, na prática, evidenciou alguns problemas. As primeiras versões desta especificação deixaram muitas áreas abertas à interpretação dos fornecedores e, como resultado, os ORBs de diferentes fornecedores não conseguiam comunicar uns com os outros, limitando a capacidade de combinação e interoperabilidade.

Para a especificação de todos os objectos envolvidos num sistema, é necessário conhecer a linguagem IDL, no entanto, dado ser de uma extrema complexidade não se apresenta como uma forte candidata para a utilização generalizada, como era inicialmente a sua vocação.

Com excepção de redes locais e corporativas onde as restrições de segurança são planeadas, a adopção do CORBA nem sempre é pacifica, pois requer o uso de portas de comunicação especiais e alterações específicas em firewalls que representam um elevado risco de segurança.

2.5.2 DCOM

O DCOM, *Distributed Component Object Model*, é o mecanismo da Microsoft para realizar chamadas a procedimentos remotamente, um modelo de programação orientado a objectos e desenhado para facilitar a interoperabilidade do software [Gordon 2000].

Neste modelo a interacção é definida por forma a que o cliente e o componente se possam conectar sem necessidade de ter qualquer recurso intermediário. Um servidor pode criar instâncias de objectos de múltiplas classes de objectos. Um objecto pode suportar múltiplas interfaces, cada uma representando uma visão ou comportamento diferente do objecto.

Tal como a maioria dos sistemas baseados em objectos, também o DCOM recorre a uma linguagem de definição das interfaces dos seus objectos. No entanto, no DCOM não houve a preocupação de definir ligações com linguagens de alto nível, facto que se revelou como um factor limitativo desta tecnologia.

A solução encontrada para resolver o problema da interoperabilidade foi o recurso a uma norma binária, proprietária da Microsoft. Qualquer linguagem que entenda esta norma, pode criar e utilizar objectos DCOM, no entanto, só poderá ser utilizado em plataformas Microsoft, o que significa que tanto o cliente como o servidor têm de ser sistemas baseados em tecnologias da Microsoft.

Com o DCOM os detalhes sobre as questões de localização dos componentes não são especificados no código. Cada componente contém um identificador único de 128 bits, denominado por *Class Identifier* (CLSID), que associa uma classe de objectos a uma Dynamic Linked Library (DLL) ou aplicação executável (EXE).

Assim, para que o cliente localize o componente, precisa apenas de conhecer o seu identificador, estando este no registo da máquina local ou em qualquer máquina da rede. Esta independência de localização do DCOM simplifica a tarefa de distribuição dos componentes de uma aplicação.

2.5.3 JavaRMI

A especificação Java RMI é a tecnologia da Sun para desenvolver aplicações distribuídas sem que estas tenham de se preocupar com as plataformas que as vão suportar, sendo apenas necessário que as mesmas suportem a *Java Virtual Machine* (JVM).

A JavaRMI utiliza o *Remote Method Protocol* (RMP) para permitir a invocação de métodos de um objecto localizado numa máquina distinta, de forma transparente. O protocolo RMP é proprietário da Sun e não foi desenhado com o objectivo de interoperar com outros ORBs ou linguagens, por este motivo o seu uso como tecnologia de suporte dos *backbones* da *Internet* é substancialmente limitado.

A interacção dos clientes JavaRMI com os objectos remotos é feita através das interfaces (classes abstractas) que são fornecidas pelo servidor, o cliente possui a interface do objecto remoto para poder referenciá-lo, enquanto que o servidor possui a implementação do mesmo.

Devido à enorme popularidade da linguagem Java, os fornecedores de software optam por suportar a JVM nas suas plataformas, são eles: Javasoft (SunOS e Solaris), Microsoft (Windows), AIX, HPUX, Linux e MacOS. Além disso, todas as plataformas que suportem a JVM têm a possibilidade de utilizar JavaRMI.

2.6 Resumo

Um dos grandes desafios para as tecnologias da informação tem sido, e continuará a ser, a integração das aplicações dentro e entre as empresas ou organizações.

Independentemente do caminho tomado pelas organizações relativamente à preferência por uma das soluções existentes no mercado ou pelo desenvolvimento de uma solução à medida, o essencial é que os projectos sejam desenhados de forma a garantir um sistema integrado, de acordo com a arquitectura do Sistema de Informação que certamente evoluirá ao longo do tempo [Amaral e Varajão 2000].

A característica da heterogeneidade nos sistemas distribuídos impõe a necessidade de especificações abertas, com interfaces padronizadas e públicas, levando ao desenvolvimento de *software* de *middleware* baseado fortemente em tecnologia e protocolos de objectos que promovam a interoperabilidade num ambiente distribuído e heterogéneo.

Este capítulo aborda diversas tecnologias de integração. Inicialmente foram introduzidas as mensagens assíncronas, em seguida, foi explicada a arquitectura dos procedimentos remotos, proposta no início dos anos 80 mas que ainda hoje representa o "estado da arte" nos sistemas distribuídos.

Apresentaram-se as tecnologias CORBA, DCOM e JavaRMI por serem nos últimos anos as tecnologias mais utilizadas para o desenvolvimento de aplicações distribuídas. No entanto, algumas limitações de cada uma destas tecnologias faz com que elas não sejam a melhor solução. Essas limitações resultam de causas como: falta de interoperabilidade, complexidade de implementação, orientação à plataforma, elevada dependência do código e comunicação tipicamente síncrona.

A utilização de padrões abertos e altamente difundidos torna a tecnologia Web Services uma excelente alternativa para a integração de SI em ambientes heterogéneos, pela sua simplicidade, versatilidade e abertura, fundamentais para a sua escolha como tecnologia preferencial de integração.

No próximo capítulo apresenta-se a nova tecnologia de integração, os Web Services, fazendo uma análise dos conceitos tecnológicos envolvidos e respectiva função. Além disso, é feita uma apreciação das principais linhas de orientação para a construção de aplicações baseadas em Web Services.

C A P Í T U L O I I I

3. Web Services

O tema central desta tese são os Web Services (WS) para implementação de aplicações distribuídas. Através da utilização de WS, aplicações em várias localizações, num ambiente global – *Internet*, podem ser facilmente integradas como se fossem parte de um único sistema.

Por forma a tornar esta interoperabilidade possível são necessários protocolos independentes dos sistemas operativos, plataformas e linguagens de desenvolvimento. Neste capítulo são analisadas em detalhe estas normas *standard* nas quais os WS se baseiam, tais como XML (*eXtensible Markup Language*), SOAP *(Simple Object Access Protocol*), UDDI (*Universal Description, Discovery and Integration*) e WSDL (*Web Services Definition Language*).

Por fim, apresenta-se o conjunto de organizações que têm como principal objectivo criar as linhas orientadoras para o desenvolvimento de WS, de forma a manter a inter-operacionalidade.

3.1 Definição de Web Service

A equipa responsável pela arquitectura de serviços da IBM define Web Service como sendo uma aplicação modular auto-descritiva, que pode ser publicada, localizada e invocada a partir de qualquer ponto de uma rede, geralmente a Web [IBM 2000].

Segundo definição do *World Wide Web Consortium* (W3C), um Web Service é um componente de software identificado por um URI[2] [W3C 2005], cujas interfaces e ligações são capazes de ser definidas, descritas e descobertas como artefactos XML. Um Web Service suporta interacções directas com outros componentes de software usando mensagens baseadas em XML, trocadas via protocolos na *Internet* [W3C 2004].

Existem provavelmente tantas definições de Web Service como existem empresas a implementar aplicações baseadas nesta arquitectura, mas todas possuem em comum as características a seguir aqui reunidas.

Os Web Services herdam dos sistemas orientados aos objectos as noções fundamentais de encapsulamento e reutilização. Os WS são componentes de software que encapsulam funcionalidade discreta, representam uma "caixa-preta" que pode ser reutilizada sem a preocupação de se saber como funciona internamente. O facto de encapsularem funcionalidade discreta determina que os WS sejam auto-contidos e executem apenas uma actividade, para a qual estão perfeitamente definidas as entradas e saídas, de modo a ser possível determinar de uma forma simples a sua funcionalidade. Po exemplo, como invocá-la e qual o resultado esperado.

O sucesso das tecnologias de informação para a *Internet* deve-se, em parte, ao facto de possuírem um nível de abstração suficientemente alto, de modo a facilitar a compatibilidade entre sistemas heterogéneos. A infra-estrutura de *Internet* na qual se baseiam os WS explora este nível de abstração, associando, ao mesmo tempo, informação semântica aos dados [Lopes e Ramalho 2005].

A ubiquidade dos protocolos da *Internet* e formato de dados, tais como o *Hypertext Transfer Protocol* (HTTP) e o *Extensible Markup Language* (XML), permite transferir informações de forma transparente através de *firewalls* e servidores *proxy*, tornando os WS componentes facilmente acessíveis por parte de qualquer aplicação [W3C 2006a].

---

[2] *Uniform Resource Identifier* [W3C 2005].

Uma das características mais relevantes neste novo paradigma que são os WS é a noção de que tudo é um serviço que tem uma interface publicada e que comunica através de mensagens. Desta forma, estes serviços podem ser implementados em qualquer plataforma e linguagem de programação, desde que possam produzir e consumir as mensagens definidas pela interface. Podem ainda ser alterados sem qualquer impacto nas aplicações clientes, desde que mantenham a mesma interface, porque são *loosly coupled*, ou seja a ligação aos WS é estabelecida dinamicamente apenas quando necessária.

Para permitir que as aplicações clientes possam descobrir de forma dinâmica a interface do Web Service, os serviços devem ser descritos utilizando a especificação WSDL, esta norma permite publicar as informações sobre a sintaxe, métodos, parâmetros e localização dos serviços [W3C 2007a]. Desta forma, para construir uma aplicação que realize chamadas para um determinado Web Service, obtém-se o ficheiro WSDL associado e com base nas informações nele contidas desenvolve-se a aplicação.

Os ficheiros WSDL podem ser armazenados no UDDI, que é uma proposta para directório geral de registo e pesquisa de serviços disponíveis. Os clientes interessados num determinado serviço podem fazer uma procura nesse directório e obter informações relativas à interface e à localização do serviço.

Nas secções seguintes faz-se uma análise pormenorizada de cada uma das normas essenciais nas arquitecturas baseadas em Web Services.

3.2 XML

EXtensible Markup Language (XML) é uma meta-linguagem utilizada para representação de informação, que se assume como norma no intercâmbio dessa informação em ambientes heterogéneos.

O organismo *World Wide Web Consortium* (W3C) definiu o XML baseando-se na linguagem *Standard Generalized Markup Language* (SGML) da norma ISO 8879 [Navarro et al. 2000], foi desenhado com o intuito de permitir a definição de documentos estruturados, auto-descritivos e legíveis − como texto, sendo

suficientemente formal para permitir a sua validação, tendo estrutura suficiente para suportar documentos complexos e grandes quantidades de informação.

Um documento XML usa uma sintaxe simples e de fácil compreensão, constituído por uma árvore de elementos relacionados, utiliza etiquetas que caracterizam os dados e a sua formatação, não se encontra limitado relativamente ao número de etiquetas que se podem definir. A figura 3 exemplifica o conteúdo de um documento XML muito simples.

```
<?xml version="1.0" ?>
<user>
      <lastname>Saraiva</lastname>
      <age>32</age>
      <dateborn>1974-08-11</dateborn>
</user>
```

Figura 3: Documento XML “bem-formado”

Do texto oficial da especificação do XML [W3C 2006b] destacam-se as seguintes características:

- Extensível e estruturado

  É permitido definir qualquer etiqueta (*tag*) que se considere necessária, utilizando elementos que caracterizem os dados e identifiquem a estrutura do documento tornando assim mais fácil a sua interpretação.

- Passível de ser validado

  É possível efectuar a validação de um documento XML verificando se respeita todas as regras de formação e se o seu conteúdo estiver de acordo com uma especificação definida previamente num documento DTD (*Document Type Declaration*) ou num XSD (XML *Schema Definition*) associado.

Um documento XML bem formado é um documento que obedece às regras de sintaxe XML [W3Schools 2007]:

- Deve começar pela declaração XML;
- Deve ter um elemento raiz único, isto é, deve existir um elemento para o qual nem a etiqueta de início nem a de fim estão dentro de qualquer outro elemento;

- Todas as etiquetas abertas devem ser fechadas;
- Deve ter em atenção que as etiquetas XML são *case-sensitive*;
- Todos os elementos devem ser aninhados correctamente, ou seja, uma etiqueta aberta dentro de um elemento deve ter a correspondente etiqueta de fim antes desse elemento se fechar;
- Todos os valores de atributos precisam de estar entre aspas;
- Devem utilizar-se entidades XML para caracteres especiais.

3.2.1 Document Type Declaration

Um DTD consiste num conjunto de declarações que especifica a estrutura hierárquica de um documento XML, descrevendo quais e de que forma as *tags* podem ser usadas.

Um DTD tem uma sintaxe própria diferente de XML, com *tags* definidas para representar os elementos usados. Na figura 4, apresenta-se o exemplo de um possível DTD para o documento XML ilustrado na figura 3.

```
<?xml version="1.0" ?>
<!ELEMENT user (lastname, age, dateborn)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT dateborn (#PCDATA)>
```

Figura 4: Um possível DTD para o XML da figura 3

Como se pode verificar este DTD define a estrutura de um utilizador (*user*). A linha 2 define que um utilizador possui três elementos: “lastname, age, dateborn”, as linhas 3 a 5 definem os seus elementos constituintes como sendo do tipo “#PCDATA”, ou seja, apenas texto.

Um DTD pode estar contido num documento XML ou pode ser declarado externamente sendo associado ao documento através de uma referência com a seguinte sintaxe: “<!DOCTYPE elemento_raiz SYSTEM “nome_do_ficheiro” >”.

3.2.2 *XML* Schema Definition

Tal como um DTD, o conceito de XML *Schema* é definir a estrutura de um documento XML. No entanto, é ele próprio escrito em XML, o que lhe permite uma maior extensibilidade, contrariamente ao DTD que possui uma sintaxe própria. Tornou-se uma recomendação do W3C em Maio de 2001, tentando solucionar algumas lacunas importantes encontradas no DTD [W3C 2007b].

Embora haja mecanismos no padrão DTD para validar a obrigatoriedade de elementos, a sua sequência, a frequência com que devem aparecer no documento, etc., não há forma de validar o tipo desses elementos. Basicamente, para as DTDs os elementos são do tipo "texto" ou são vazios, para o exemplo apresentado é possível especificar a existência de um elemento chamado "dateborn", mas não é possível assegurar que o conteúdo desse elemento seja uma data válida [W3C 2007c].

A figura 5 mostra um possível XML *Schema* para o exemplo que se apresentou na figura 3, note-se que, tal como em todos os XSD, ele contém o elemento raiz "schema" onde se inclui a informação a respeito do namespace.

Tal como se pode observar, nas restantes linhas, define-se o elemento "user" como sendo do tipo complexo (*complexType*), porque contém outros elementos, sendo constituído pela sequência (*sequence*) de elementos simples "lastname, age, dateborn".

```
<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:complexType name="user">
            <xs:sequence>
                  <xs:element name="lastname" type="xs:string" />
                  <xs:element name="age" type="xs:integer" />
                  <xs:element name="dateborn" type="xs:date" />
            </xs:sequence>
      </xs:complexType>
</xs:schema>
```

Figura 5: Um possível XML *Schema* para o XML da figura 3

Um elemento simples não pode conter outros elementos, o seu conteúdo deve pertencer aos denominados tipos de dados base, nestes encontram-se a grande generalidade dos tipos primitivos das mais diversas linguagens de programação: *string*,

boolean, *float*, *double*, *integer*, *decimal*, *binary*, *date*, *datetime*, etc. [Ramalho e Henriques 2002].

A extensibilidade dos XML *Schema* permite criar novos tipos de dados personalizados, a partir dos tipos base, dando origem aos tipos simples derivados. Neste tema do XML a extensibilidade nos XML *Schemas* é bastante abrangente, no entanto não se pretende no âmbito desta tese alongar mais o assunto.

3.3 SOAP

O *Simple Object Access Protocol* (SOAP) é uma norma que especifica a estrutura de mensagens utilizadas para que as aplicações comuniquem entre si, garantindo que os intervenientes, emissor e receptor, mesmo que de sistemas heterogéneos, consigam interpretar as mensagens e conheçam os seus detalhes.

Inicialmente os principais contributos para a criação do SOAP vieram da Microsoft, DevelopMentor e UserLand Software, mais tarde contaram também com o envolvimento da IBM e de muitas outras empresas agrupadas num consórcio denominado *Word Wide Web Consortium* (W3C).

A especificação do SOAP resulta de mais de dois anos de desenvolvimento que culminaram com a submissão da versão 1.1 ao W3C, o qual a aceitou como ponto de partida para futuras normalizações. A actual versão 1.2 foi um trabalho de normalização que decorreu sob os auspícios do XML *Protocol Working Group do W3C* [W3C 2007d].

A norma SOAP consiste em quatro partes: um envelope para transmissão de mensagens que descreve os vários aspectos que serão utilizados para descrever a mensagem; um conjunto de directrizes de codificação para expressar os tipos de dados definidos ao nível das aplicações; uma convenção para representar as invocações a procedimentos remotos (RPC) e respectivas respostas; e ainda, uma *framework* para a troca de envelopes entre aplicações utilizando um protocolo de transporte.

O SOAP, protocolo de comunicação pode, potencialmente, ser utilizado em combinação com uma variedade de outros protocolos, como protocolos de transporte HTTP, SMTP, FTP e suporta o mecanismo clássico de RPC, que consiste na troca de mensagens síncronas de pedido, seguido de resposta. Adicionalmente, suporta outros mecanismos de comunicação, nomeadamente: mensagens num único sentido, sem resposta (i.e., *one-way*); notificação (resposta tipo "push"); ou publicação/subscrição.

De modo a ser um protocolo independente da heterogeneidade de sistemas, foi necessário fazer algumas concessões com o objectivo de reduzir o nível da sua complexidade, todavia a sua especificação contém algumas limitações, como o facto de não dispor de uma forma padrão de autenticação e segurança.

No entanto, os autores da especificação SOAP argumentam que esses problemas podem ser facilmente ultrapassados através da utilização do protocolo HTTPS para codificação dos métodos já preconizados para autenticação em termos de HTTP. Além disso, para fazer face a estas e outras necessidades surgem módulos, SOAP *Extensions*, que se podem associar ao cabeçalho das mensagens [Smith 2007]. Na secção seguinte apresentam-se algumas destas extensões.

3.3.1 *SOAP* Extensions

As extensões SOAP são mecanismos de extensibilidade para mensagens, utilizados para oferecer maior integridade e confidencialidade às aplicações baseadas em Web Services. Através de SOAP *Extensions* é possível analisar ou modificar uma mensagem em qualquer altura do seu ciclo de vida entre o cliente e o servidor [MSDN 2007].

WS-Security é uma extensão SOAP que descreve a forma como usar as assinaturas digitais e a cifragem nas mensagens, de forma a garantir a sua integridade, autenticidade e confidencialidade. Os WS podem usar esta extensão para, através da análise às mensagens e aos *tokens* de segurança que recebem, decidirem se vão aceitar ou rejeitar o pedido [Chappell 2003].

WS-Reliability é uma especificação que permite adicionar fiabilidade à comunicação com SOAP independentemente do tipo de transporte utilizado, em aspectos como a garantia de entrega das mensagens, a eliminação de mensagens duplicadas e a ordem correcta de chegada das mensagens [Iwasa 2004].

XML *Digital Signatures* resulta de um esforço conjunto entre W3C e IETF (*Internet Engineering Task Force*) para definir uma norma *standard* de representar assinaturas digitais num documento XML, com o intuito de garantir a integridade dos dados, a autenticação do cliente e suporte à não-repudiação dos dados assinados [W3C 2002a].

XML *Encryption* é um protocolo desenvolvido sob a orientação do W3C e que pretende descrever a forma de cifrar e decifrar conteúdos digitais, incluindo documentos XML ou secções destes. O XML *Encryption* inclui uma sintaxe XML para representar o conteúdo cifrado, porém não se preocupa com detalhes de segurança mais amplos, como assinaturas, autenticações e autorizações [W3C 2002b].

Security Assertion Markup Language (SAML) é uma norma que torna possível a troca de informação relativa à autenticação e autorização entre aplicações, permitindo que uma aplicação cliente apenas se autentique uma única vez, independentemente dos diversos serviços a que pretenda aceder [OASIS 2007b].

SOAP *Messages With Attachments* (SWA) possibilita enviar um ou vários documentos anexados à mensagem SOAP numa estrutura MIME, *Multipurpose Internet Mail Extensions*. Estes documentos que não se encontram em formato XML, podem ser, por exemplo, documentos Word, PDF, imagens em formato GIF ou JPEG, ou mesmo assinaturas digitais [OASIS 2006].

3.3.2 Mensagem SOAP

Uma mensagem SOAP é uma mensagem escrita em XML, composta por um envelope que tipicamente contém um cabeçalho e um corpo, apresenta-se de seguida a figura 6 para ilustrar esta estrutura [MSDN 2007].

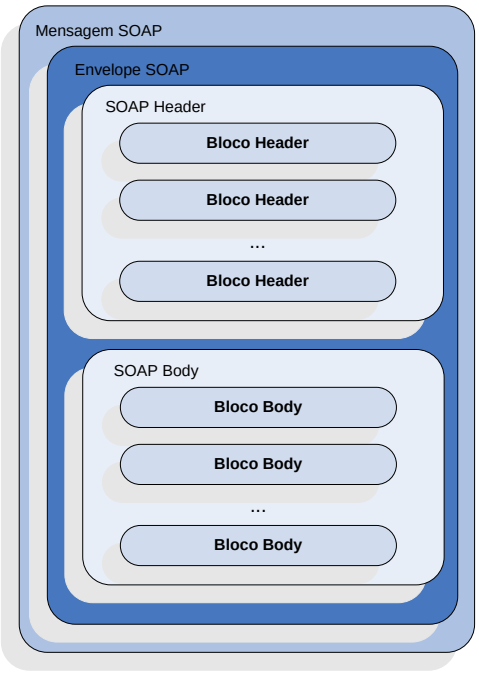

Figura 6: Representação de uma mensagem SOAP

O elemento envelope é obrigatório e é o elemento principal de uma mensagem SOAP, representa a raiz da estrutura XML dentro da qual todos os outros elementos da mensagem estão encapsulados.

O envelope contém um cabeçalho opcional que, quando presente, contém informações relativas à segurança, encaminhamento ou destinatário da mensagem. Uma mensagem pode conter zero ou mais blocos de cabeçalho, que serão embebidos no cabeçalho.

O corpo da mensagem é definido pelo elemento *Body*, é um elemento obrigatório, deve estar localizado imediatamente depois do elemento *Header*, caso ele exista, e contém a informação da mensagem escrita em XML dirigida ao receptor final.

A figura 7 apresenta um exemplo simples de uma mensagem SOAP para transmitir através do protocolo HTTP. É efectuado um pedido de execução no corpo de um POST HTTP, este invoca o método *GetMunicipe* e o resultado é obtido através da resposta a este pedido HTTP, a resposta é apresentada na figura 8.

Apesar de simples, este exemplo ilustra os principais elementos de uma mensagem SOAP. O envelope SOAP, representado pelo elemento *soap:Envelope*

contém apenas o corpo da mensagem, representado por *soap:Body*, os elementos opcionais *soap:Header* e *soap:Fault* não estão presentes neste caso.

```
POST /WSACentral/wsMunicipe.asmx HTTP/1.1
Host: ws.shortcut.pt
Content-Type:text/xml; charset=utf-8
Content-Length: 250
SOAPAction:"http://ws.shortcut.pt/ws.WSAutarquiasCentral/GetMunicipe"
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
      <soap:Body xmlns:m=" http://ws.shortcut.pt/ws.WSAutarquiasCentral">">
            <m:GetMunicipe>
                  <m:cod_autarquia>123</m:cod_autarquia>
                  <m:num_contribuinte>123456789</m:num_contribuinte>
            </m:GetMunicipe>
      </soap:Body>
</soap:Envelope>
```

Figura 7: Exemplo de pedido SOAP sobre HTTP

```
HTTP/1.1 200 OK
Content-Type:text/xml; charset=utf-8
Content-Length: 250
<?xml version="1.0"?>
  <soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Body xmlns:m="http://ws.shortcut.pt/ws.WSAutarquiasCentral">
      <m:GetMunicipeResponse>
        <m:Contrib>123456789</m:Contrib>
        <m:Tipo_User>0</m:Tipo_User>
      </m:GetMunicipeResponse>
    </soap:Body>
  </soap:Envelope>
```

Figura 8: Exemplo de resposta SOAP sobre HTTP

O *namespace* associado ao envelope SOAP é "http://www.w3.org/2001/12/soap-envelope" e é indicado em todos os elementos através do prefixo *soap*, do mesmo modo, o conteúdo do elemento *soap:Body* contém um documento XML em que todos os seus elementos são classificados com o *namespace* respectivo. Os *namespaces* são utilizados para evitar conflitos de documentos XML que apresentam os mesmos nomes para descrever diferentes elementos.

Um cliente HTTP efectua uma conexão a um servidor HTTP utilizando TCP, após estabelecer essa ligação, o cliente pode enviar uma mensagem de pedido para o servidor. Um pedido HTTP POST especifica pelo menos dois cabeçalhos: *Content-Type*

e *Content-Length*. O primeiro define o tipo MIME para a mensagem (*application/soap*) e a codificação de caracteres utilizada no corpo do pedido/resposta (*utf-8*); o segundo especifica o número de *bytes* no corpo do pedido/resposta (250 *bytes*).

Nos casos em que HTTP é o protocolo de transporte a utilizar, o protocolo SOAP define um cabeçalho HTTP identificado por *SOAPAction*, cujo valor é um URI que indica o objectivo da mensagem, no exemplo apresentado, figura 7, *SOAPAction* contém o valor "http://ws.shortcut.pt/ws.WSAutarquiasCentral/GetMunicipe", que indica que se pretende invocar o método *GetMunicipe* do Web Service.

Após receber o pedido, o servidor processa-o e envia a resposta para o cliente, essa resposta contém um código que indica o estado do pedido, 200 indica sucesso, se o servidor não conseguisse descodificar o pedido a resposta teria devolvido: HTTP/1.1 400 *BAD REQUEST* e *Content-Length*: 0.

A indicação sobre o estado do pedido é representada sempre por um número inteiro, onde valores entre 200 e 299 indicam sucesso. No caso da mensagem conter um elemento *soap:Fault*, elemento que representa um erro, a especificação obriga a que a informação de estado devolva o valor 500, indicando um *Internal Server Error*.

3.4 UDDI

O projecto *Universal Description, Discovery and Integration* (UDDI) foi lançado pelas empresas IBM, Microsoft e Ariba, mas actualmente é liderado por um consórcio de empresas muito mais abrangente designado por *Organization for the Advancement of Structured Information Standards* (OASIS) [OASIS 2007a].

O UDDI é uma iniciativa que pretende criar uma infra-estrutura global e independente das plataformas de desenvolvimento, capaz de permitir a localização e obtenção de informação comercial e técnica relativa aos Web Services disponíveis, de maneira mais fácil rápida e dinâmica.

Com efeito, o UDDI corresponde a um repositório de nomes de empresas ou outras organizações e de descrição dos serviços, suportado por uma base de dados

distribuída de Web Services. Conceptualmente, as informações estão disponíveis em três componentes, descritas como:

- *White Pages* (páginas brancas): contêm informações acerca das organizações fornecedoras de WS, tais como nome, morada, contactos e descrição da organização;

- *Yellow Pages* (páginas amarelas): classificam os WS em categorias nas quais os seus fornecedores se inserem, como por exemplo, tipo de serviço ou localização geográfica;

- *Green Pages* (páginas verdes): constituem o conjunto de informações técnicas sobre os WS disponibilizados, bem como informações relativas à sua localização.

O modelo de dados do UDDI é definido por um XML *Schema* que contém quatro tipos principais de informação, utilizados para descrever na prática os três tipos de informação acima mencionados, são eles:

- *BusinessEntity*: este elemento representa toda a informação conhecida de uma organização que publica Web Services, correspondendo à informação das *White Pages*;

- *BusinessService*: contém a informação relativa a um determinado WS publicado no UDDI, sendo que cada elemento *businessService* pertence a um único *businessEntity*. Esta informação concretiza portanto o conceito das *Yellow Pages*;

- *BindingTemplate*: nesta estrutura encontram-se as descrições técnicas sobre o WS registado e descreve-se como a estrutura *businessService* utiliza várias informações técnicas, correspondendo à informação encontrada nas Green Pages;

- *TModel*: este elemento é utilizado para indicar a localização de informação técnica adicional de um WS como, por exemplo, uma descrição que se encontre num ficheiro externo.

O UDDI providencia três funções principais, conhecidas como publicação, descoberta e ligação. A publicação torna possível que uma organização divulgue os seus serviços, a descoberta permite que o cliente, ou invocador do serviço, procure e encontre um determinado serviço, e finalmente a ligação para que o cliente possa estabelecer a interacção com o serviço.

Os servidores de UDDI podem ser públicos, acedidos via *Internet*, ou privados, acedidos, por exemplo, em *Intranets* de empresas. Os processos de registo e invocação ao UDDI podem ser feitos através de uma interface programática, utilizando uma API ou Web Services disponibilizados para o efeito, ou por pessoas através de uma interface.

3.5 WSDL

A especificação *Web Service Description Language* (WSDL) foi inicialmente uma proposta da IBM, Microsoft e cerca de uma dúzia de outras empresas, no entanto, tal como as normas XML e SOAP foi aceite pelo W3C e está agora sob a orientação do *XML Protocol Working Group*.

Um documento WSDL é escrito utilizando XML e pretende descrever a interface de um ou vários Web Services, apresentando os métodos possíveis de invocar em cada serviço e os respectivos parâmetros que eles aceitam, os tipos de dados, os protocolos que podem ser utilizados e o endereço da sua localização [W3C 2007a].

O WSDL é extensível para permitir a descrição de serviços e mensagens, independentemente dos formatos de mensagens e dos protocolos de rede que sejam usados, possibilitando assim que um serviço seja descrito de forma totalmente transparente da sua implementação.

Cada documento WSDL é constituído por duas partes, como mostra a figura 9, a primeira visa a definição dos serviços em termos abstractos e, a segunda, tem como objectivo concretizar a definição abstracta em termos de protocolos e transportes suportados.

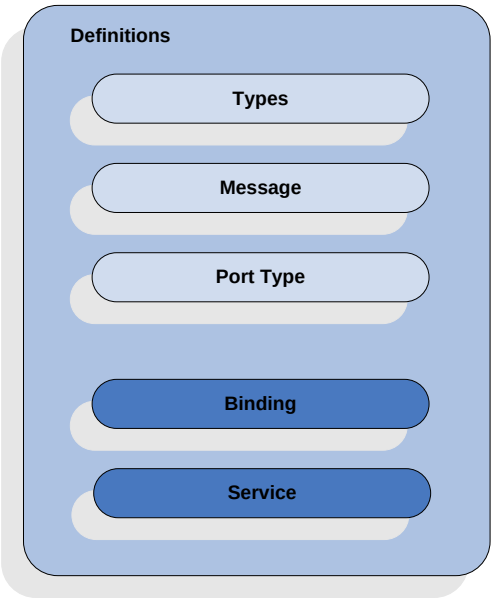

Figura 9: Representação de um documento WSDL

Os principais elementos que constituem a definição da interface de um Web Service num documento WSDL são:

- *Definitions*: é o elemento raiz do formato XML do documento, sendo aqui que se definem os *namespaces* que são utilizados ao longo do documento;

- *Types*: este elemento define os tipos de dados utilizados nas mensagens, ou seja no elemento *message*. Esta definição deverá ser feita segundo o XML *Schema Definition*;

- *Message*: define as assinaturas de entrada e saída de cada método. Cada elemento mensagem definido refere-se a um pedido ou a uma resposta e é constituído por um mais elementos do tipo *part*;

- *Port Type:* identifica as operações (métodos) expostos por um WS. A cada operação, definida pelo elemento *operation*, estão associados um pedido e uma resposta identificados respectivamente pelos elementos *input* e *output*.

Os elementos que pertencem à definição da interface do serviços são completamente independentes do protocolo de transporte. No entanto, para a concretização do serviço, torna-se necessário associar um protocolo específico a cada definição abstracta dada pelo elemento *Port Type*. Os elementos que permitem esta associação são:

- *Binding:* este elemento especifica qual o protocolo concreto para determinada operação. A especificação WSDL permite ligações com protocolos e formatos de mensagem, como SOAP, HTTP GET /POST e MIME;

- *Service:* no seu elemento *port* identifica-se o endereço físico onde se localiza o WS, através de um URI. Cada elemento *port* está sempre associado a um elemento *binding*.

Na figura 10 apresenta-se a estrutura genérica de um documento WSDL, neste exemplo ilustra-se a utilização dos principais elementos apresentados anteriormente.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http=http://schemas.xmlsoap.org/wsdl/http/
             xmlns:soap=http://schemas.xmlsoap.org/wsdl/soap/
             xmlns:s=http://www.w3.org/2001/XMLSchema
             xmlns:s0=http://ws.shortcut.pt/ws.WSAutarquiasCentral/
             xmlns:soapenc=http://schemas.xmlsoap.org/soap/encoding/
             targetNamespace=http://ws.shortcut.pt/ws.WSAutarquiasCentral/
             xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
  <s:schema elementFormDefault="qualified"
            targetNamespace=http://ws.shortcut.pt/ws.WSAutarquiasCentral/>
    <s:element name="Authenticate">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="client" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="pass" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="AuthenticateResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="AuthenticateResult"
                     type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
</types>
<message name="AuthenticateSoapIn">
```

```
  <part name="parameters" element="s0:Authenticate" />
</message>
<message name="AuthenticateSoapOut">
  <part name="parameters" element="s0:AuthenticateResponse" />
</message>
<portType name="WSAAuthenticationSoap">
  <operation name="Authenticate">
    <input message="s0:AuthenticateSoapIn" />
    <output message="s0:AuthenticateSoapOut" />
  </operation>
</portType>
<binding name="WSAAuthenticationSoap" type="s0:WSAAuthenticationSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="Authenticate">
    <soap:operation soapAction=http://ws.shortcut.pt/ws.WSAutarquiasCentral/Authenticate
                    style="document"/>
    <input>
      <soap:body use="literal" />
      <soap:header message="s0:AuthenticateSHWSClientCredentials"
                   part="SHWSClientCredentials" use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
<service name="WSAAuthentication">
  <documentation>WSAAuthentication no Host Central</documentation>
  <port name="WSAAuthenticationSoap" binding="s0:WSAAuthenticationSoap">
    <soap:address location="http://ws.shortcut.pt/WSACentral/wsAAuthentication.asmx" />
  </port>
</service>
</definitions>
```

Figura 10: Estrutura genérica de um documento WSDL

3.6 Interoperabilidade

Para os utilizadores finais de um sistema, a diferença entre integração e interoperabilidade é geralmente irrelevante. Efectivamente, o facto de várias aplicações funcionarem em conjunto de modo a responderem aos requisitos de um negócio é o aspecto mais valorizado por este tipo de utilizadores [Gardner 2003].

No entanto, em termos tecnológicos, as diferenças são importantes, as tendências actuais em TI vão no sentido de automatizar a integração mais profundamente, ao nível da plataforma e com base em normas *standard*. Deste modo, promove-se a interoperabilidade entre sistemas, no sentido em que são colocados a comunicar e interagir, independentemente da abordagem tecnológica de suporte.

Esta oportunidade de interoperabilidade surge dos padrões e especificações de Web Services e do papel que se espera deles como catalizadores de uma nova arquitectura de paradigma de aplicações orientados para serviços.

As principais empresas concordaram que a definição de um conjunto de especificações seria de grande interesse tanto para os utilizadores como para elas próprias e começaram a cooperar inseridas em consórcios sem fins lucrativos. As organizações mais significativas no contexto geral da *Internet* e em particular dos Web Services são: o *World Wide Web Consortium* (W3C), a *Organization for the Advancement of Structured Information Standards* (OASIS), a *Web Services Interoperability Organization* (WS-I) e a *Internet Engineering Task Force* (IETF).

O W3C tem como objectivo desenvolver tecnologias interoperáveis, especificações e ferramentas para potenciar a *World Wide Web*, além disso, dispõe de um fórum de informações, comunicação e aprendizagem destas tecnologias [W3C 2007e].

O W3C define as especificações dos protocolos HTTP, XML, XML Schema, SOAP, WSDL, XSLT, XML *Encryption* e XML *Signature*, entre outras, e ainda, estabelece a definição formal dos Web Services no documento *WS-Architecture* [W3C 2004].

A OASIS promove o desenvolvimento, a convergência e a adopção de normas para o comércio electrónico. Controla as especificações de UDDI, WS-Security, WS-Reliability, SAML e ebXML, entre outras, no *site* desta organização podem encontrar-se todas as informações acerca destas tecnologias [OASIS 2007a].

A IETF é uma comunidade internacional de grande dimensão que junta engenheiros de redes, operadores, fabricantes e investigadores preocupados com a evolução da arquitectura da *Internet*. As normas reguladas por esta comunidade são: HTTP, URI (URL e URN), TCP e IP, entre outras [IETF 2007].

A IEFT produz os documentos *Requests for Comments* (RFC), que são notas técnicas e organizacionais sobre a *Internet*. Estas notas discutem muitos aspectos de redes de computadores, incluindo protocolos, procedimentos, programas e conceitos, bem como notas de reuniões e opiniões.

A WS-I define-se como sendo uma organização cuja missão é promover a interoperabilidade de Web Services, entre diferentes plataformas, aplicações e

linguagens de programação [WS-I 2007a]. A figura 11 representa o papel desempenhado pela WS-I enquanto intermediário entre as normas *standard* e a indústria.

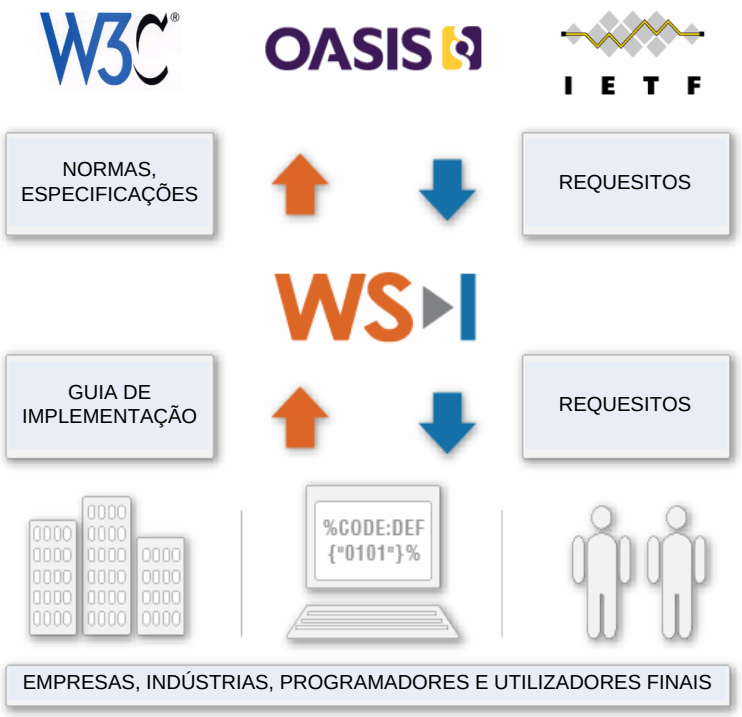

Figura 11: WS-I, *standards* e indústria [WS-I 2005]

A WS-I define os perfis de interoperabilidade. Cada perfil abrange um conjunto de normas sobre um aspecto específico de uma solução e fornece orientações para a sua implementação, disponibiliza casos de utilização, aplicações de exemplo e testes para aferir o cumprimento da especificação.

Actualmente existe um perfil de base denominado WS-I *Basic Profile* 1.2 que cobre as áreas de mensagem (HTTP, SOAP 1.2, MTOM, WS-Addressing 1.0), descrição (WSDL 1.1), descoberta (UDDI 3.0) [WS-I 2007b] e os perfis de segurança, o WS-I *Basic Security Profile* [WS-I 2007c] e o WS-I *Reliable Secure Profile* 1.0 [WS-I 2006].

O *SOAPBuilders Interoperability Lab* fornece uma plataforma, disponível na *Internet*, para testar a interoperabilidade com mais de 30 produtos que suportam SOAP. Os resultados dos testes são depois disponibilizados pelos fabricantes dos produtos [SOAP 2003].

3.7 Resumo

Neste capítulo abordamos os conceitos e características da plataforma de Web Services, apresentando e descrevendo as normas *standard* em que eles se baseiam.

Conceptualmente, os Web Services representam uma nova forma de implementar aplicações distribuídas, aproveitando a infra-estrutura e a flexibilidade que a *Internet* oferece. Os WS tornaram-se assim na tecnologia mais significativa dos sistemas distribuídos, alcançaram a independência dos dados introduzida pelo XML e permitem resolver os problemas de integração entre empresas ou entre diferentes plataformas.

Os protocolos em que os WS se baseiam definem essencialmente os aspectos relacionados com a troca de mensagens entre aplicações, ou seja, entre a aplicação cliente e o Web Service, permitem descrever a interface do WS e como eles podem ser descobertos dinamicamente pelas aplicações cliente.

O funcionamento dos Web Services é baseado no modelo de arquitectura orientado a serviços e utiliza a infra-estrutura da *Internet* bem como as suas normas, da seguinte forma:

- Um formato normalizado de representar os dados – XML;
- Uma infra-estrutura de comunicação global – TCP/IP;
- Um protocolo de transporte de dados normalizado – HTTP;
- Um formato de mensagens comum – SOAP;
- Uma linguagem de descrição dos serviços – WSDL;
- Um modo de descobrir serviços e fornecedores de serviços na Web – UDDI.

Os aspectos relacionados com a segurança, gestão de transacções e encaminhamento de mensagens estão também alinhados com os princípios técnicos fundamentais, nomeadamente, a composição de protocolos e a interoperabilidade baseada em normas. As mensagens SOAP são extensíveis, permitindo incorporar informação de segurança em cabeçalhos específicos para esse efeito.

C A P Í T U L O I V

4. Proposta de um Sistema de Informação usando Web Services

Após a reunião dos conceitos fundamentais de integração de SI, tendo essas finalidades e características por referência, apresenta-se, neste capítulo, uma arquitectura de um sistema de integração utilizando Web Services.

Numa primeira parte define-se um modelo conceptual, apresentam-se os seus elementos constituintes, as suas interacções e o papel de cada um deles na integração.

De seguida, propõe-se e implementa-se um sistema funcional através da integração entre os sistemas TRAS-OS-MONTES.NET e WSAutarquias. Inclui-se uma discussão do sistema implementado e detalha-se de forma pormenorizada a sua implementação, sendo feita uma descrição funcional e técnica de cada componente.

No final é apresentada uma breve súmula da implementação do sistema e dos temas abordados, são ainda tecidas algumas conclusões.

4.1 Modelo conceptual

Os processos de negócio entre empresas têm sido tradicionalmente caracterizados por não apresentarem qualquer integração entre os seus respectivos SI, consequentemente, resultam no consumo intensivo de recursos humanos e em vários erros de inconsistência ou duplicação de informação.

Constata-se que este modelo não satisfaz adequadamente as interacções entre e dentro das empresas. No entanto, mesmo desejando a integração, a maioria das empresas não tem interesse em se desfazer dos seus actuais SI, pelo que surge a necessidade de encontrar uma solução que permita aos diferentes sistemas comunicar entre si.

A elaboração de um modelo de arquitectura para a integração de SI, envolvendo e suportando o estabelecimento de múltiplas relações de colaboração de uma forma aberta, distribuída e dinâmica entre distintos parceiros, deve contribuir para que as empresas [Gardner 2003]:

- Aproveitem melhor as aplicações e as plataformas que possuem;
- Evoluam as aplicações antigas para interfaces, aplicações e arquitecturas mais actuais;
- Generalizem os tipos de conteúdos e os distribuam de forma eficiente de forma a alcançar um maior número de dispositivos e utilizadores finais;
- Desenvolvam serviços interoperáveis de modo a criar novo valor para as aplicações;
- Migrem para plataformas e aplicações mais recentes que facilitem a modernização e consolidação;
- Façam com que o lado comercial da organização sinta que a TI é uma fonte de produtividade e não apenas uma fonte de gastos imprevisíveis e sempre crescentes.

Em suma, as finalidades globais e características desejadas da integração devem contribuir para que as empresas aumentem a eficiência dos sistemas, possibilitando o rápido alcance dos objectivos e agilizando os seus processos de negócio.

No entanto, existem muitas dificuldades tecnológicas de interoperação entre SI, não só por serem total ou parcialmente suportados por diferentes infra-estruturas aplicacionais, possuírem formatos de dados específicos e "*front-ends*" proprietários, mas também pela segurança entre os sistemas que terá que ser garantida.

Tendo em conta estas dificuldades, os principais fabricantes de *software* têm vindo a especificar normas técnicas apropriadas, *standards* para garantir a interoperabilidade entre SI, que embora não tenham atingido completamente a maturidade, podem considerar-se já ubíquos.

Neste contexto de normalização, aparece um dos mais recentes paradigmas de desenvolvimento de *software*, que dá pelo nome de Web Services, que preconiza uma arquitectura de sistemas de software designada por "arquitectura orientada aos serviços", SOA (*Service Oriented Architecture*), e permite criar camadas tecnológicas integradoras para os SI.

Uma arquitectura baseada em Web Services possibilita a criação de um nível de abstração mais elevado entre aplicações e na definição das interfaces dos serviços, dando maior flexibilidade e independência na integração de SI,

4.2 Modelo de arquitectura

Para construir aplicações baseadas no modelo SOA existe um certo número de requisitos que devem ser preenchidos, concretamente: a integração de recursos de software deve ser *loosely coupled*, a comunicação entre aplicações deve ser compatível com os standards da *Internet* e a definição e documentação da interface dos serviços dos recursos de software deve estar publicamente acessível.

Para o desenvolvimento de aplicações que preencham os requisitos enumerados anteriormente é necessário que exista uma infra-estrutura tecnológica, que permita a criação, publicação, identificação e activação dos serviços [Snell *et al.* 2001].

Como se pode constatar pela figura 12, a arquitectura orientada aos serviços consiste na interacção entre dois intervenientes principais, um desempenhando o papel de invocador ou consumidor do serviço: *Service Consumer*; o outro, o papel de fornecedor do serviço: *Service Provider*. O terceiro interveniente: *Service Registry*, desempenha o papel de mediador ou intermediário na descoberta entre consumidor e fornecedor e consequentemente promove o estabelecimento da interacção.

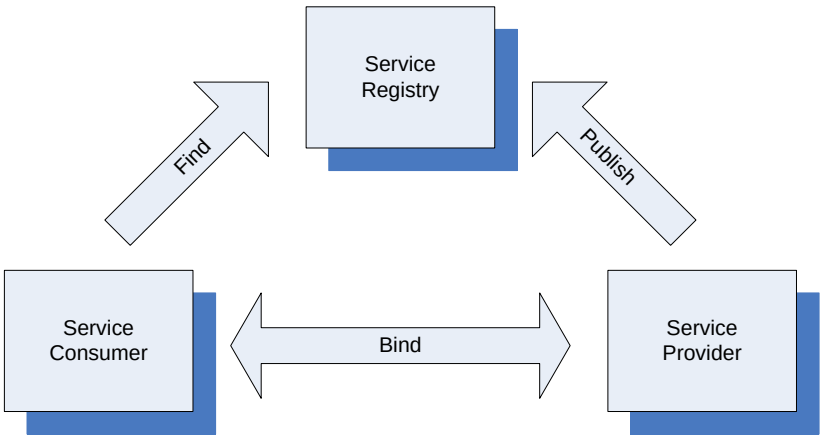

Figura 12: Arquitectura orientada aos serviços

Pelo facto dos três participantes no modelo SOA: o *Service Consumer*, o *Service Provider* e o *Service Registry*, terem um papel fundamental em todo o funcionamento dessa infra-estrutura, vale a pena serem descritos em mais de detalhe.

- *Service Provider*

  É o nó na rede, *Internet* ou *Intranet*, que providencia o acesso à interface de um serviço de software de uma funcionalidade específica. O *Service Provider* disponibiliza o acesso a serviços de um sistema empresarial, um subsistema ou um componente.

- *Service Consumer*

  É o nó na rede que se liga a um serviço de um *Service Provider* e o usa para implementar uma solução de negócio. Num modelo de arquitectura orientada ao serviço o *Service Consumer* não é uma aplicação, no entanto, pode considerar-se como a aplicação cliente no nó.

- *Service Registry*

  É o nó na rede que é o repositório da descrição dos serviços, contém um registo global dos serviços disponíveis podendo ser usado como um livro de endereços para procurar a localização dos serviços. O *Service Consumer* interroga o *Service Registry* para localizar o requerido *Service Provider* e o próprio serviço em si.

O modelo SOA foi desenhado para permitir que os nós supracitados realizem transacções entre si, ou seja, interactuem para implementar as três operações básicas seguintes:

- *Publish services*

  Os *Services Providers* publicam os seus serviços no *Service Registry*. A informação publicada inclui a definição da interface do serviço, a localização do *Service Provider* e se possível outra informação de suporte ou documentação.

- *Find service*

  Os *Services Consumers* procuram os serviços requeridos ou desejados usando o *Service Registry*.

- *Bind to services*

  Os *Services Consumers* ligam-se a um serviço específico providenciado pelo *Service Provider*, sendo que, este processo de ligação inclui autenticação dos consumidores.

Ambas as operações "*find service*" e "*bind to service*" devem poder ser realizadas dinamicamente, permitindo que as aplicações se configurem a si próprias. Este é um aspecto importante, uma vez que os serviços não se destinam a ser usados directamente pelo utilizador. O método de operação baseia-se em trocas de dados ao nível programático, executadas por outros componentes ou aplicações. Por exemplo, se uma aplicação que está a tentar utilizar um serviço, se apercebe que o tempo de resposta do *Service Provider* se torna inaceitável, então ela deve poder decidir mudar para outro *Service Provider* em *run-time*.

Os três nós definidos numa arquitectura orientada ao serviço são mapeados nos elementos de uma solução baseada em WS, apresentada na figura 13. O modelo foi desenhado tendo em conta estes princípios e mostra como duas aplicações podem comunicar entre si e criar uma interacção entre os seus processos.

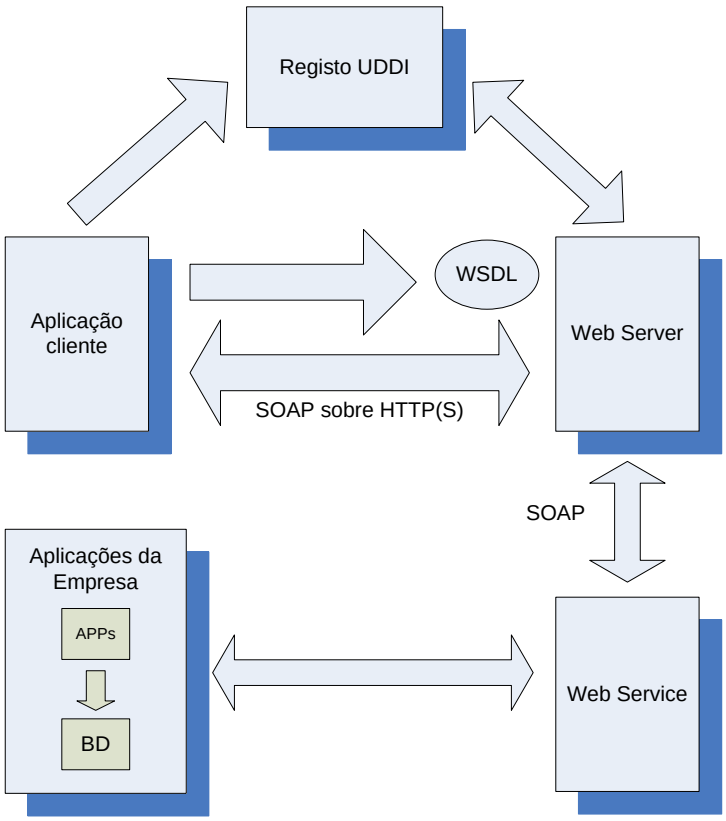

Figura 13: Arquitectura baseada em Web Services

Um Web Service é composto por dois elementos físicos: o serviço e a descrição do serviço. O serviço é a implementação da sua respectiva interface. Consiste num módulo de software instalado numa plataforma computacional com acesso à rede e oferecido pelo *Service Provider*. Um serviço pode ser invocado, ou pode ele próprio funcionar como um invocador de outro serviço.

A descrição do serviço contém os detalhes da interface e da implementação de um serviço, o que inclui os tipos de dados, operações, informação de ligação (*binding*) e localização de rede. Pode ainda incluir metadados e informação de categorização para facilitar as actividades de descoberta e utilização por invocadores do serviço.

O conceito de descrição do serviço é fundamental para permitir que a arquitectura minimize as dependências explícitas e directas entre os seus componentes. O objectivo é que nem o invocador, nem o fornecedor de um serviço tenham de conhecer os seus detalhes internos, como, por exemplo, plataforma de suporte ou linguagem de programação, para poderem interagir entre si.

A descrição do serviço pode ser publicada num registo UDDI, para tornar o respectivo serviço conhecido num determinado contexto. O padrão que permite descrever os WS segundo um formato XML é o WSDL, uma especificação normalizada sob as directivas do W3C [Booth *et al.* 2004].

Considerando as dificuldades encontradas por outras arquitecturas de objectos distribuídos, em chegar a um acordo sobre um formato binário suficientemente compatível com um vasto conjunto de sistemas, o SOAP apresenta-se como o padrão adequado para o empacotamento e formatação de mensagens a transaccionar.

O protocolo SOAP não obriga a nenhum meio de transporte específico, podendo ser usado qualquer um dos protocolos estabelecidos na *Internet* ou mesmo fora dela. Contudo, o principal interesse é a possibilidade de se poder efectuar a chamada a procedimento remotos (RPC) sobre HTTP ou HTTPS, sendo esta aliás a situação descrita na especificação W3C [Booth *et al.* 2004].

A ideia fundamental de um modelo baseado em WS é que estes devem ser componentes distribuídos, facilmente acessíveis através dos protocolos padrão da *Internet* como o HTTP. Este aspecto dos WS é o que lhes permite atravessar de uma forma transparente as políticas de *firewalls* das empresas, tornando o seu acesso por parte de qualquer aplicação praticamente ubíquo.

Para a arquitectura das aplicações, a organização mais lógica que tem prevalecido é uma arquitectura de três camadas [Trowbridge 2004]. Ela é constituída por uma camada de apresentação, que contém somente a formatação e o controle de dados necessários para a interface com o utilizador, uma camada de aplicação, que contém a lógica do sistema que controla a sequência das operações, aplica as regras de negócio da aplicação e garante a integridade referencial, e uma camada de dados, que contém as funcionalidades básicas de CRUD (*create*, *read*, *update* e *delete*) ou seja, inserção, leitura, actualização e remoção no repositório de dados da aplicação.

Numa arquitectura direccionada a serviços, naturalmente nem todos os objectos são candidatos naturais a serem WS. É necessário definir e representar quais os objectos

que devem ser expostos dessa forma e quais devem fazer parte da camada que contém a funcionalidade da aplicação em si [Boar 2003].

Assim, define-se que o modelo mais apropriado ao desenvolvimento deste tipo de aplicações deve ter seis camadas, como apresentado de seguida [Promon 2002].

- Camada de apresentação

  A camada de apresentação permite ao utilizador interagir com os serviços disponibilizados pelo sistema. Esta camada contempla os diversos componentes directamente relacionados com a interface, é dependente do dispositivo cliente, uma vez que contém os elementos finais de apresentação.

- Camada de lógica da apresentação

  A manipulação da apresentação dos elementos visuais que compõem a aplicação é da responsabilidade da camada de lógica de apresentação. É nesta camada que são preparadas as informações a serem exibidas de acordo com o dispositivo cliente, considerando factores como a ordem desses elementos e a validação das informações dadas pelas regras de negócio.

- Camada de serviço da aplicação

  Esta camada disponibiliza uma interface para ser usada por outras aplicações, é nela que são desenvolvidos componentes seguindo os conceitos de Web Services. Esses componentes aceitam pedidos de clientes usando protocolos padrão da *Internet*, como o HTTP, permitindo a sua utilização por outros sistemas.

- Camada de lógica de negócio

  Esta é a camada que implementa toda a lógica de negócio, isto é, comunica com a camada de persistência para recolher os dados e efectua o tratamento dos mesmos de acordo com as regras de negócio especificadas. Representa o núcleo da aplicação, em que diversas considerações e regras estão protegidas da intervenção das restantes camadas.

- Camada de persistência

  A camada de persistência propõe a abstracção do acesso aos dados armazenados no Sistema Gestor de Base de Dados (SGBD) para a camada de lógica de negócio.

- Camada de SGBD

  Esta camada lógica refere-se à base de dados em termos físicos, representa o SGBD que armazena os dados que dizem respeito à aplicação.

A figura 14 ilustra como, conceptualmente, dois sistemas desenvolvidos numa arquitectura de WS podem comunicar entre si, interagindo através da interface implementada na camada de serviço da aplicação.

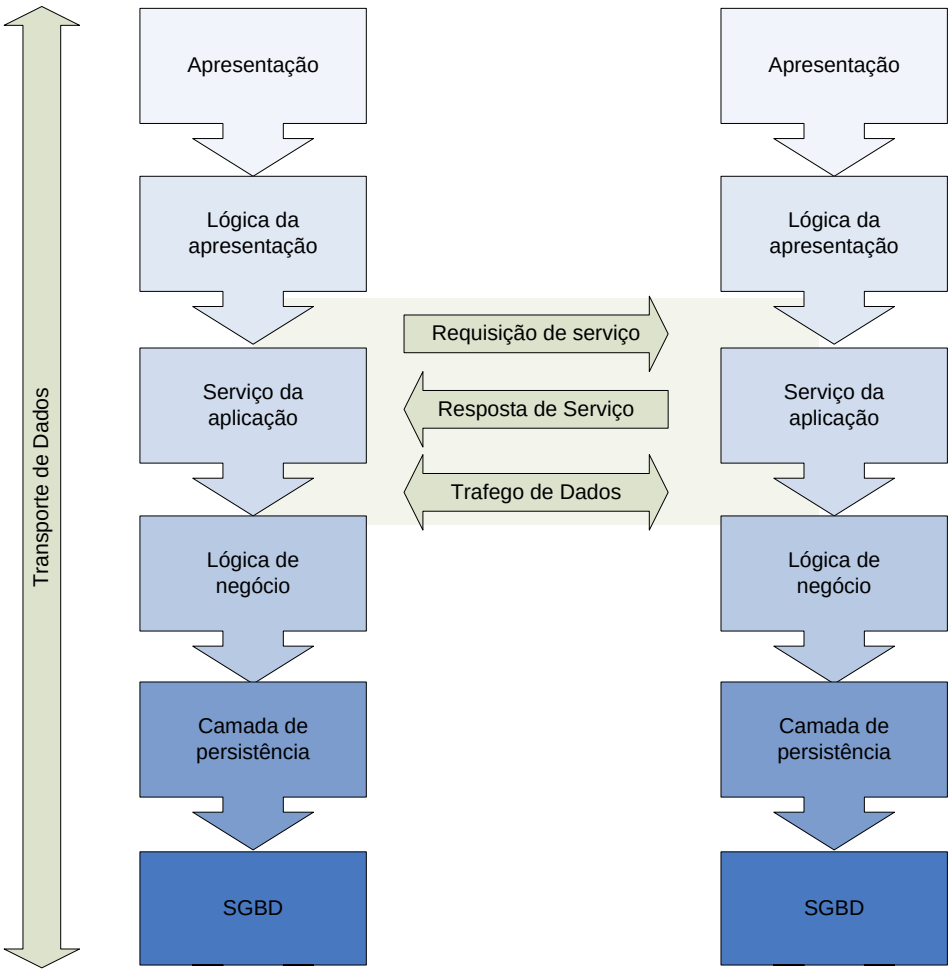

Figura 14: Modelo de seis camadas de uma arquitectura usando Web Services

Ao longo de vários anos, foram desenvolvidas várias soluções que pretendiam possibilitar que uma aplicação utilizasse uma interface de acesso a uma outra aplicação. Entre elas podemos citar DCE RPC, DCOM, CORBA, RMI em Java, tal como foi referido no segundo capítulo.

Apesar destas tecnologias terem basicamente a mesma intenção funcional, isto é, desenvolver os seus programas de forma a utilizar aplicações remotas como se estivessem acessíveis no seu próprio ambiente, tinham uma série de problemas e limitações. Além disso a interoperabilidade entre plataformas prometida, era muito complexa, semanticamente ambígua e requeria um esforço de desenvolvimento enorme.

Assim, o modelo de WS baseado numa linguagem padrão, o XML, representa uma boa alternativa para responder aos requisitos básicos de comunicação e compatibilização de dados necessários à interoperabilidade de aplicações, que podem, ou não, estar dentro da mesma empresa.

4.3 Modelo tecnológico

A tecnologia subjacente aos WS baseia-se fortemente na *Internet*, através de um conjunto de standards. Este conjunto tecnológico está agrupado por camadas que desempenham o seu papel num domínio de actuação limitado e perfeitamente definido.

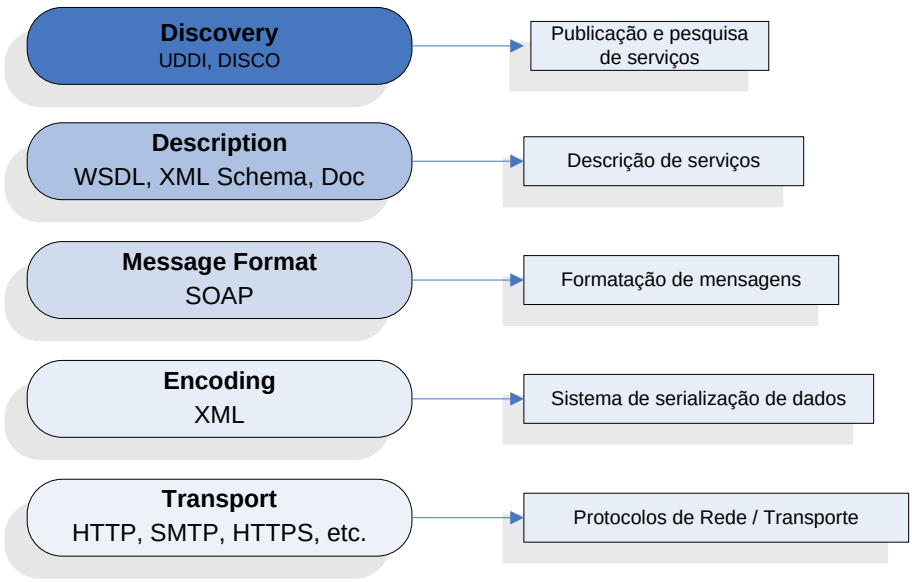

Figura 15: Suporte tecnológico de uma arquitectura baseada em Web Services [Short 2002]

Na figura 15 é apresentada a sobreposição das camadas referidas e a sua relação com as principais tecnologias. Cada uma delas será descrita de seguida:

- Publicação e pesquisa de serviços

  Uma especificação que permite expor as funcionalidades de registo, manutenção e pesquisa de informação é o *Universal Description, Discovery and Integration* (UDDI). O UDDI pretende providenciar locais electrónicos onde as aplicações cliente irão à procura dos serviços necessários.

  Além disso, o UDDI providencia várias opções para a descoberta e a ligação aos WS. Na sua forma mais simples, o UDDI permite que um invocador se ligue a um WS em tempo de desenho, pelo conhecimento prévio da sua respectiva descrição do serviço e endereço. Outra solução, passa pela ligação dinâmica a um serviço estático, isto é, cuja descrição do serviço também é pré-conhecida, mas o seu endereço é obtido dinamicamente em tempo de execução.

  Finalmente, a solução mais versátil, passa pelo invocador descobrir em tempo de execução a descrição do serviço e o endereço do WS que pode satisfazer os seus requisitos.

- Descrição de serviços

  Uma forma de descrever as mensagens possíveis de invocar num determinado serviço e de saber quais são os respectivos parâmetros que elas aceitam é o programador obter a documentação relativa ao serviço e incluir no seu código esse conhecimento. No entanto, esta não é provavelmente a melhor maneira de se utilizarem os WS, mas para casos pontuais é uma solução.

  A melhor forma é, haver uma maneira de se definirem, de modo padrão e abstracto, as características de um serviço, de modo que dinamicamente se possa aceder à sua funcionalidade sem necessidade de grandes conhecimentos prévios.

  A *Web Services Description Language* (WSDL) pretende atingir precisamente este propósito [Christensen *et al.* 2001]. Uma linguagem baseada em XML que serve para descrição de serviços em rede, definindo-os como colecções de pontos de contacto ou portas.

- Formatação de mensagens

  Na tentativa de resolver os problemas existentes ao estabelecer a comunicação entre aplicações distribuídas, num ambiente heterogéneo, foi criado o *Simple Object Access Protocol* (SOAP). O protocolo SOAP prescreve um mecanismo simples e leve para a troca de informação estruturada e tipificada entre pares num ambiente descentralizado e distribuído usando XML [Gudgin *et al.* 2003]. Na prática, o SOAP é uma sintaxe XML para a troca de mensagens e, sendo baseado em XML, é independente quer da linguagem quer da plataforma.

  O SOAP não define ou impõe qualquer semântica aplicacional, quer seja um modelo de programação, quer seja semântica específica de implementação. Este aspecto é de enorme importância, pois permite que quer o serviço, quer o invocador do serviço, sejam aplicações desenvolvidas sobre diferentes tecnologias.

- Sistema de serialização de dados

  O que se pretende com a serialização de dados é representar a informação produzida por programas, normalmente num formato binário ou de texto com uma estrutura proprietária e formato ambíguo, num formato extensível e independente da plataforma [W3C 2006c].

  Embora os dados XML sejam representados sobre a forma de texto não pretendem ser lidos por pessoas, mas sim pelo próprio computador, no entanto a sua representação permite que, por exemplo, um programador consiga inspeccionar facilmente um ficheiro XML e corrigir um problema, usando apenas um editor de texto e não necessitando do programa original que o criou.

- Protocolos de rede / transporte

  A tecnologia subjacente aos Web Services baseia-se fortemente na *Internet* e na respectiva tecnologia e standards. Esta opção prende-se com a necessidade de tirar partido das infra-estruturas já criadas e dos protocolos já tornados ubíquos pela *Internet*.

  O uso do HTTP, o mesmo protocolo usado pelos servidores Web e pelos browsers, apresenta múltiplas vantagens já que é de simples utilização, é bem

tolerado pelas *firewalls* e conta com uma panóplia enorme de adereços, como o balanceamento, *caching*, *proxy*, etc.

Para dados mais sensíveis pode ser usado o protocolo HTTPS, que adiciona cifragem ao HTTP. Para serviços que não se baseiem em pergunta/reposta imediata ou que necessitem de funcionalidades de filas de mensagens, encontra-se disponível o protocolo SMTP.

De notar que os WS não implicam um protocolo específico, podendo utilizar para o seu funcionamento qualquer um dos protocolos padrão da *Internet*. A sua associação com o protocolo HTTP resulta de razões de flexibilidade e reutilização de infra-estruturas.

4.4 Apresentação do projecto

O projecto de integração objecto desta dissertação nasce da necessidade de proporcionar um conjunto de serviços de diversas entidades da rede de administração pública, de forma integrada num portal designado TRAS-OS-MONTES.NET.

O portal TRAS-OS-MONTES.NET disponibiliza ao cidadão em geral e em qualquer local com acesso à *Internet*, a possibilidade de aceder a diversos serviços. Estes serviços estavam até então disponíveis apenas através da presença física de uma pessoa no local de atendimento da própria entidade ou nos gabinetes de apoio ao cidadão (GAC) criados no âmbito do programa "Cidades e Regiões Digitais".

Além disso, é objectivo deste projecto ir mais longe. Pretende-se que a consulta dos processos de um cidadão dentro do portal, seja resultado de uma consulta directa aos SI das entidades parceiras.

Até agora, a informação disponível no TRAS-OS-MONTES.NET era apenas a dos pedidos efectuados a partir do portal ou dos gabinetes de apoio ao cidadão. Esses pedidos eram consultados e actualizados manualmente por um agente intermediário da instituição a quem o pedido se destina.

O protótipo de integração desenvolve-se no âmbito de uma parceria entre a UTAD e as empresas Medidata [Medidata 2007] e Shorcut [Shortcut 2007]. Estas empresas implementam os sistemas utilizados nas autarquias com os quais se pretende que o portal TRAS-OS-MONTES.NET interaja.

Nesse sentido, a interacção desejada insere-se na categoria de SI distribuídos, aproveitando a infra-estrutura que a *Internet* disponibiliza e é construída segundo o modelo de arquitectura orientada a serviços, através da tecnologia de WS. Na figura 16 pode observar-se um diagrama que representa uma visão geral da integração definida.

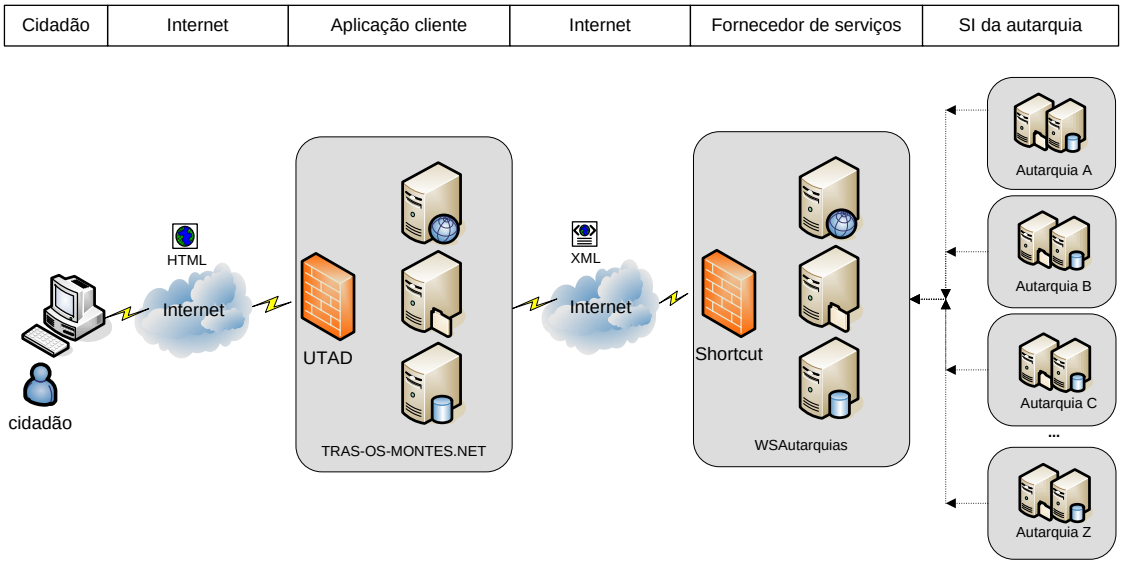

Figura 16: Visão geral do protótipo de integração

A integração é conseguida através da criação de duas novas camadas de software, a primeira colocada na aplicação cliente TRAS-OS-MONTES.NET e a segunda sobre o SI das autarquias, permitindo assim, a total integração das aplicações entre a entidade requisitante e as entidades parceiras fornecedoras da informação.

O sistema WSAutarquias é constituído pelos WS que consultam uma réplica parcial das bases de dados das autarquias. Esta réplica é conseguida através da sincronização periódica com os sistemas locais existentes nas próprias instituições. Portanto, a informação consultável no portal TRAS-OS-MONTES.NET é obtida em tempo real por interacção com o sistema WSAutarquias e refere-se ao conteúdo da base de dados replicada das autarquias.

4.5 Protótipo de integração

Nesta secção procede-se à concretização do modelo de integração de SI apresentado nas secções anteriores. Efectua-se a descrição funcional e técnica dos componentes da solução, bem como do conjunto de processos que permitem que um utilizador do portal TRAS-OS-MONTES.NET interaja com um sistema distinto, o WSAutarquias, com a transparência, fiabilidade e segurança exigidas.

Mesmo sendo uma solução de integração restrita no contexto dos SI das Administrações Municipais, estamos convictos dos benefícios do presente trabalho. Efectivamente, a capacidade da plataforma se integrar no futuro com outros sistemas requererá um esforço muito menor, uma vez que é possível a reutilização de componentes desenvolvidos.

A relevância deste trabalho é justificada pelo seu alinhamento com as directrizes da Administração Pública, de pesquisar novas tecnologias que fomentem a integração de sistemas. Pretende-se utilizar princípios e técnicas ligadas à interoperabilidade com o intuito de se obter uma solução versátil, sendo facilmente adaptada no que diz respeito à incorporação de novos serviços de entidades distintas.

Na figura 17 apresenta-se o modelo de entidades básicas e as respectivas associações na solução implementada, são elas:

- *User*, a entidade capaz de interagir com o sistema, personifica um cidadão que registado no portal;
- *Group*, um conjunto de utilizadores, que têm em comum as mesmas regras de privilégios dentro do sistema;
- *Role*, define os privilégios de um utilizador ou conjunto de utilizadores englobados num grupo, sobre as entidades do sistema;
- *Permission*, está associada a *Role* e habilita os privilégios de uma regra específica a um utilizador num dado contexto, para executar acções sobre entidades do sistema;

- *Context*, é a entidade principal do sistema e suporta as acções que os utilizadores podem efectuar no sistema. Agrega elementos como, por exemplo, actividades e serviços para um utilizador que tem definida uma regra específica;
- *Activity*, uma actividade descreve uma acção ou um conjunto de acções com um objectivo específico;
- *Service*, esta entidade representa um recurso que pode ser usado num contexto ou numa actividade com a finalidade de fornecer um serviço. Este, tanto pode ser um serviço local como um serviço remoto.

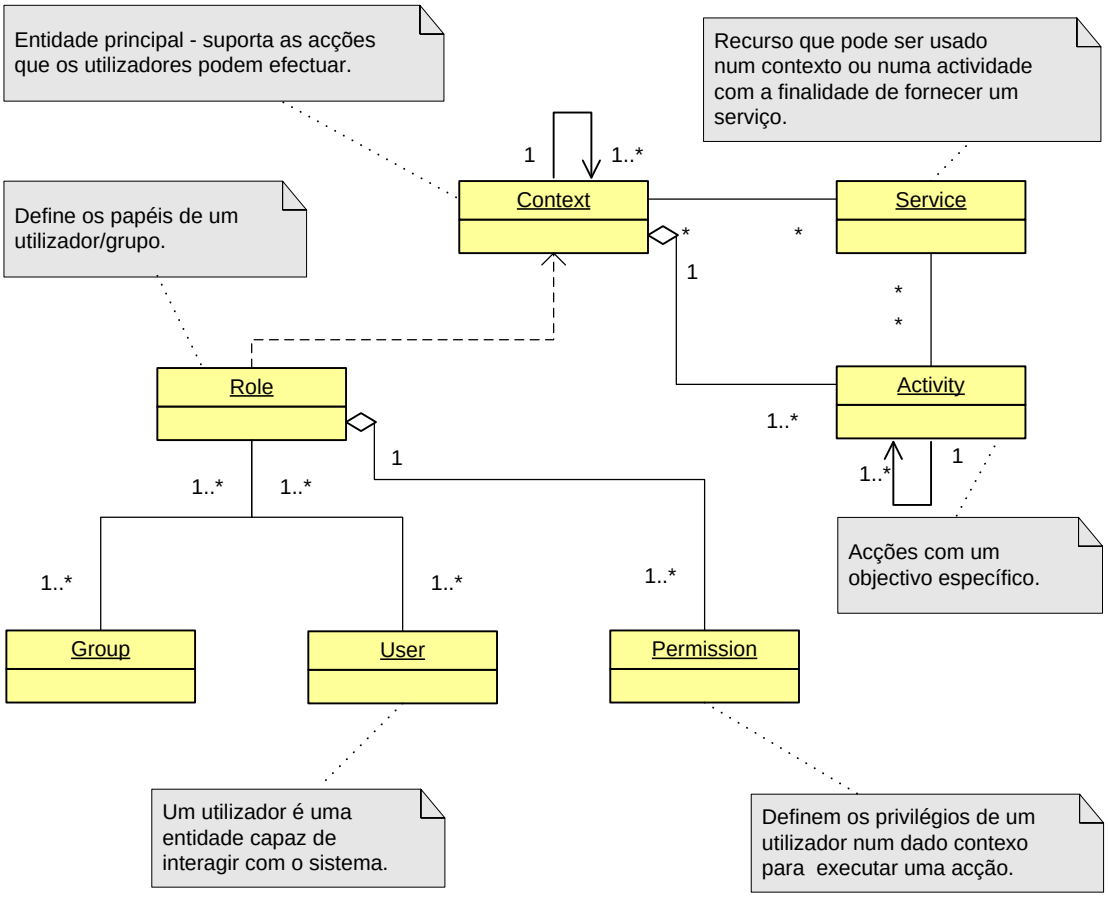
Entidade principal - suporta as acções que os utilizadores podem efectuar.
Recurso que pode ser usado num contexto ou numa actividade com a finalidade de fornecer um serviço.
Define os papéis de um utilizador/grupo.
Context
Service
Role
Activity
Group
User
Permission
Acções com um objectivo específico.
Um utilizador é uma entidade capaz de interagir com o sistema.
Definem os privilégios de um utilizador num dado contexo para executar uma acção.

Figura 17: Modelo de entidades do protótipo de integração

Pretende-se que o portal TRAS-OS-MONTES.NET seja uma plataforma de integração que disponibilize aos cidadãos em geral, a possibilidade de consultar as requisições de serviços efectuadas à Administração Pública, mais especificamente a

uma Autarquia, de uma forma totalmente transparente. Ou seja, o utilizador poder aceder a processos que estão em sistemas remotos sem se aperceber.

4.5.1 Plataforma tecnológica

O portal TRAS-OS-MONTES.NET é uma aplicação desenvolvida sobre produtos da *Microsoft*. O ambiente de programação utilizado foi *Microsoft Visual Studio .NET 2003* e as linguagens o C# e o VB.NET.

Esta implementação assenta na plataforma *Microsoft .NET*, mais concretamente na *Framework .NET v1.1*, a correr sobre o sistema operativo *Microsoft Windows 2003 Server*, que implementa um servidor *Web* através do *Internet Information Services* 6.0.

O Sistema de Gestão de Bases de Dados que suporta o portal TRAS-OS-MONTES.NET e permite validar os seus utilizadores encontra-se implementado em *SQL Server 2000 Enterprise Edition*. O acesso a esta base de dados utiliza a segurança integrada do *Windows*, seguindo as recomendações e padrões da Microsoft para o cenário em causa.

Uma vez validada a informação de *login* do utilizador no portal TRAS-OS-MONTES.NET, e a sua intenção em consultar a informação disponível remotamente no sistema WSAutarquias, é realizada uma nova autenticação entre os dois sistemas, desta vez transparente ao utilizador. A informação trocada via HTTP ou HTTPS e formato SOAP, é previamente cifrada utilizando o componente de segurança *SHSecurityClient.dll v1.1.4322*, desenvolvido pela empresa Shortcut no âmbito deste projecto.

O componente de segurança deve ser registado na plataforma *.Net Famework v1.1*. Além disso, este componente tem dependências do *J# .NET Redistribute Pack 1.1* que também deve estar instalado na mesma máquina. Finalmente, deve ser adicionada uma referência para este componente sempre que se que pretenda usar a referida segurança.

4.5.2 Serviços e Componentes

A arquitectura detalhada da solução implementada está ilustrada na figura 18, onde se apresentam os componentes fundamentais da solução final, resultado da integração entre o portal TRAS-OS-MONTES.NET e o sistema WSAutarquias.

O modelo é constituído por múltiplas camadas, sendo possível identificar com clareza quais as fronteiras entre cada um dos componentes ou conjunto de componentes, bem como a forma de interacção entre eles. Sublinha-se a existência de algum nível de independência entre as camadas, conferindo maior flexibilidade e reutilização de funcionalidades.

No repositório de dados do portal TRAS-OS-MONTES.NET que acenta sobre *Microsoft SQL Server 2000* encontra-se a definição dos privilégios de cada um dos utilizadores do portal e dos serviços que é possível utilizar no portal.

De modo a tirar o máximo partido da plataforma de base de dados, utilizaram-se *Stored Procedures*, procedimentos codificados em linguagem T-SQL que ficam armazenados e pré-compilados na base de dados. A gestão destes artefactos é efectuada pelo *SQL Server*, nomeadamente no que respeita às permissões de acesso, optimização e armazenamento.

O acesso à base de dados foi implementado com a tecnologia ADO.NET, utilizando o seu modelo desconectado devido à natureza da aplicação ser baseada na *Web*, ficando assim garantida a optimização do acesso e correspondente *pool* de conexões.

Dado o isolamento que todos os outros blocos têm relativamente ao repositório de dados, este pode ser facilmente substituído por outro qualquer, nomeadamente por um documento XML ou por uma outra base de dados relacional como o *Oracle*. Isto é possível porque existe uma camada de persistência, representada na figura 18 por *DB Component*.

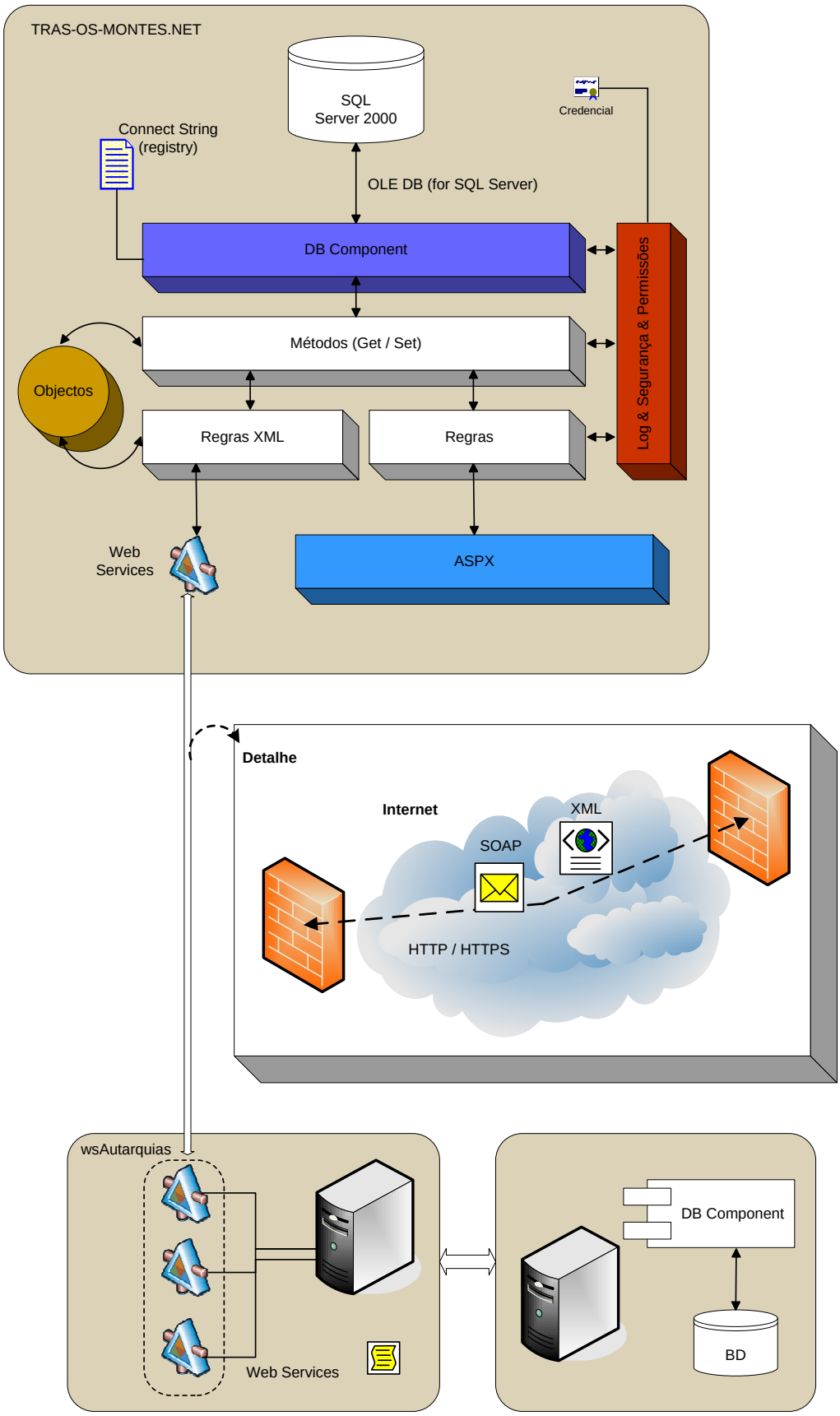

Figura 18: Arquitectura do protótipo de integração

As camadas intermédias incorporam a lógica aplicacional com as suas regras e procedimentos, de modo a fornecer uma plataforma que permita simplificar e unificar diferentes tipos de aplicações e interfaces. Várias tecnologias de middleware são utilizadas para potenciar a comunicação entre sistemas, permitindo vários níveis de integração.

As regras de formatação da informação a disponibilizar para o exterior foram implementadas em dois blocos distintos, um bloco para a comunicação com WS onde é necessário formatar as mensagens para XML, outro para utilizar com a aplicação *Web* de *front-end*. Estes blocos foram desenvolvidos de uma forma modular por forma a poderem evoluir, acomodando novas lógicas de negócios para outros canais.

A camada de interface corresponde à área de apresentação da aplicação e da informação. Através de um cliente *Web*, um *browser*, o utilizador estabelece a interacção com a aplicação *Web* TRAS-OS-MONTES.NET. Outros componentes de interface são os WS do sistema remoto WSAutarquias, estes disponibilizam informação ao portal, que é considerado o cliente desta interacção.

Os serviços do sistema WSAutarquias encontram-se disponíveis para o portal sobe a forma de *Web Methods*, alguns destes estão listados na figura 19 e o documento completo WSDL que define as mensagem de *input* e *output*, bem como as respectivas operações e a sua localização na rede será descrito no Anexo I.

Todos os métodos retornam a informação numa estrutura complexa do *.Net, System.Data.DataSet,* esta incluí, além da informação solicitada, a tabela "Erros", com o número de itens devolvidos em caso de sucesso, ou, em caso de erro, um número negativo indicativo do erro ocorrido de acordo com o serviço correspondente.

Para a apresentação foram desenvolvidas páginas ASPX utilizando tecnologias de construção de interfaces Web: VB.NET, XML, HTML, CSS e JavaScript. Este desenvolvimento foi efectuado recorrendo à ferramenta *Visual Studio .NET*.

[WebMethod] public DataSet Autentica (string Cod_autarquia, string Num_contribuinte, string Password)
Método para autenticação de um Munícipe, devolve dados relativos aos seus acessos e indicando se a sua conta ainda se encontra válida.
[WebMethod] public DataSet GetMunicipe (string Cod_autarquia, string Num_contribuinte)
Método para devolver os dados de um munícipe como nome, morada, etc.
[WebMethod] public DataSet LimparMunicipe (string Num_contribuinte)
Método para limpar toda a informação relacionada com o Munícipe da cache. Este método deverá ser chamado sempre que um munícipe abandona a sessão web.
[WebMethod] public DataSet GetProcessos (string Cod_autarquia, string Num_contribuinte)
Método para devolver todos os processos de um municipe numa dada autarquia.
[WebMethod] public DataSet GetProcesso (string Cod_autarquia, string Num_contribuinte, string IDProcesso, string IDTipoProc)
Método para devolver os dados de um processo indicando o seu número e tipo.
[WebMethod] public DataSet GetRequerimentos (string Cod_autarquia, string Num_contribuinte, string IDProcesso, string IDTipoProc)
Método para devolver os requerimentos de um processo de um munícipe.
[WebMethod] public DataSet GetRequerimento (string Cod_autarquia, string Num_contribuinte, string IDProcesso, string IDTipoProc, string IDrequerimento)
Método para devolver os dados de um requerimento.
[WebMethod] public DataSet GetCirculacoes (string Cod_autarquia, string Num_contribuinte, string IDProcesso, string IDTipoProc)
Método para devolver as circulações de um processo de um Munícipe.
[WebMethod] public DataSet GetLicencas (string Cod_autarquia, string Num_contribuinte, string IDProcesso, string IDTipoProc)
Método para devolver todas as licenças de um de um Munícipe ou todas as licenças de um processo quando este é especificado.
[WebMethod] public DataSet GetPareceresGlobal (string Cod_autarquia, string Num_contribuinte, string IDProcesso, string IDTipoProc, string IDrequerimento)
Método para devolver todos os pareceres de um determinado requerimento ou processo.

Figura 19: Web Methods de WSAutarquias

Esta ferramenta cria toda a estrutura necessária à construção de páginas Web e emprega a solução designada por *code behind*, que consiste em criar uma página ASPX que apenas contém uma directiva que indica onde se encontra localizado o código, em que linguagem está escrito e qual a classe a ser usada, pode ver-se um exemplo desta directiva na figura 20.

```
<%@ Page Language="vb" Codebehind="webServicos.aspx.vb" Inherits="wsAutarquiasTMD.webServicos" %>
```

Figura 20 : Directiva ASPX para code behind

O componente de autenticação é transversal a todas as camadas e permite que os utilizadores sejam identificados e autenticados pelo sistema de uma forma segura. Assim, foi necessário separar as funcionalidades que apenas necessitam de autenticação na base de dados local das que iriam necessitar de informações do servidor remoto.

A autenticação do portal TRAS-OS-MONTES.NET no servidor remoto é apresentada na secção 4.5.3 “Sistema de Segurança no Cliente”, onde se explica como utilizar o componente de segurança através de HTTP e HTTPS.

4.5.3 Sistema de segurança no cliente

O sistema de segurança no cliente é composto por um componente de segurança, o *SHSecurityClient.dll*, que é usado para cifrar o *username* e a *password* (dados que identificam o sistema TRAS-OS-MONTES.NET) com uma credencial fornecida. A credencial fornecida, depois de autenticada, é usada para acesso aos Web Services enquanto se encontrar válida. Foi definida para cada credencial autenticada uma durabilidade máxima de trinta minutos.

A figura 21 ilustra os procedimentos correctos a usar, para implementar segurança no acesso aos WS. Este diagrama de fluxo da credencial de sessão é explicado nos seguintes passos.

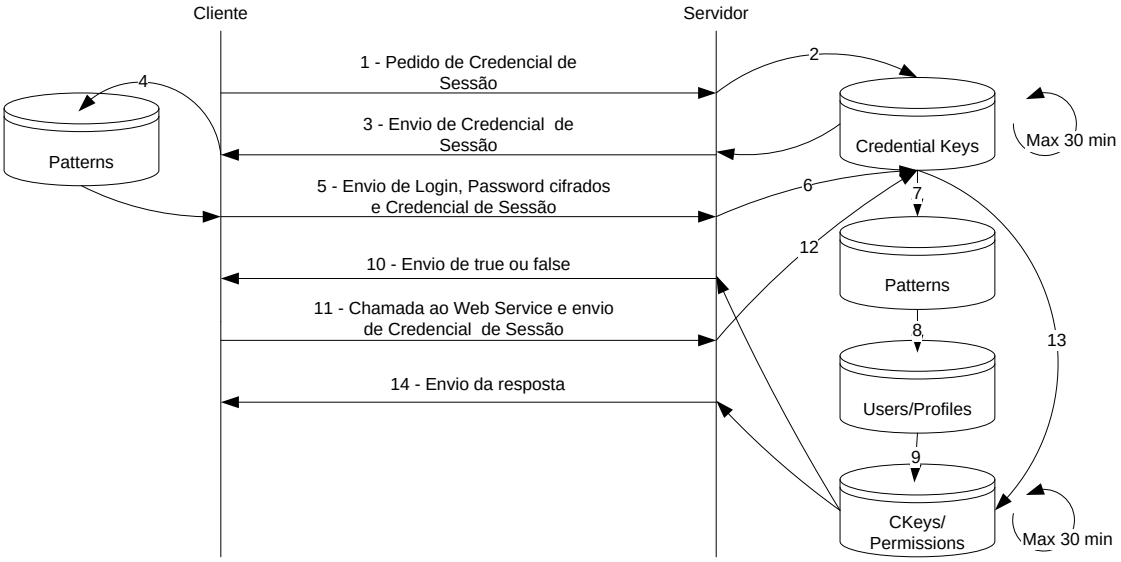

Figura 21: Diagrama de sequência de segurança [Shortcut 2003]

1. O cliente (a aplicação TRAS-OS-MONTES.NET) solicita uma credencial de sessão válida ao servidor (WSAutarquias) usando um Web Service.

2. O servidor gera uma chave de sessão única (aleatória), cifra essa chave, usa-a como sendo a credencial de segurança, insere essa credencial na base de dados de credenciais.

3. Envia a credencial de sessão ao cliente.

4. O cliente usa a credencial de sessão e o cliente de segurança *SHSecurityClient.dll* instalado no cliente para cifrar os dados de *login* e *password*.

5. O cliente efectua um pedido de autenticação com *login* e *password* cifrados usando um Web Service. A credencial de sessão também é anexada ao pedido.

6. O servidor verifica se a credencial de sessão ainda é válida, se for executa o passo 7, caso contrário envia uma SOAP *Exception*.

7. O servidor decifra a credencial de sessão.

8. O servidor decifra o *login* e a *password*, verifica se esta conta é válida e quais as suas permissões.

9. As permissões do cliente são associadas à credencial de sessão e armazenadas.

10. É devolvido ao cliente um true ou false (resultado) acerca da sua autenticação.

11. O Cliente tenta aceder a um qualquer Web Service para requisitar um serviço. A credencial de sessão também é anexada ao pedido.

12. O servidor verifica se a credencial de sessão ainda é válida, se for executa o passo 13, caso contrário envia uma SOAP *Exception*.

13. É verificado se a credencial de sessão tem permissões para aceder ao Web Service em questão.

14. É enviada a resposta do pedido ao cliente.

Sempre que ocorra uma não conformidade é enviada ao cliente uma SOAP *Exception* cuja a tabela é apresentada na tabela da figura 22. O código da utilização desta componente de segurança encontra-se no anexo II.

Soap Exception Cod name	Mensagem
SoapException.ClientFaultCode	Não foi definido o SHWSCredentialId!
SoapException.VersionMismatchFaultCode	A SHWSCredentialId usada não é válida, peça uma nova!
SoapException.ServerFaultCode	Não tem permissões para aceder a este Web Service!
SoapException.ServerFaultCod	Não tem permissões para aceder a esta Web Function!

Figura 22: Lista das SOAP *Exceptions* de WSAutarquias

A implementação desta componente de segurança sobre *Secure Socket Layer* (SSL) também é possível e recomenda-se, pois aumenta ainda mais o nível de segurança. O protocolo SSL é actualmente aceite como um mecanismo seguro para garantir integridade em comunicações baseadas em HTTP.

Para utilizar SSL é necessário que exista um certificado digital, autenticado por uma entidade certificadora (CA), no servidor que aloja os Web Services, garantindo

assim a autenticidade do servidor. Além disso, deve configurar-se o servidor para exigir o uso de SSL a todos os pedidos de clientes, o que obriga a que todas as invocações sejam realizadas por HTTPS.

É fundamental que a máquina cliente esteja configurada para confiar na autoridade de certificação que emitiu o certificado de segurança, que o instale e o use em todas as chamadas que estabelecer aos Web Services.

As chamadas aos Web Services, quer através da forma síncrona, quer através da forma assíncrona, exemplifica-se no anexo III. Nele apresenta-se o código para consultar processos utilizando o método *GetProcessos* do Web Service Processos. Quanto à utilização dos restantes serviços ela é semelhante, razão pela qual não é apresentada.

4.6 Verificação do protótipo

Este trabalho procurou dar uma contribuição para o estudo e prática desta tecnologia emergente e recheada de potencialidades. A sua importância é bem visível quando as organizações precisam que duas aplicações interoperem entre si. De seguida verifica-se a forma como este sistema foi implementado no projecto piloto.

O portal TRAS-OS-MONTES.NET é de acesso condicionado a utilizadores registados, o processo de adesão é explicado no link “Adesão” de www.tras-os-montes.net. Na figura 23 apresenta-se a página de entrada do portal, nela pode visualizar-se o preenchimento dos dados “Nº de utilizador” e “Palavra passe” exigidos para entrada.

Address
http://www.tras-os-montes.net/tmd/paginas/webentrada.aspx
trás-os-montes.net
serviços de atendimento electrónico
Página inicial + Adesão + Ajuda + Contactos + Atendimento Electrónico
Adicionar aos favoritos + Espigueiro.pt
bem-vindo
Trás-os-montes.net é o portal de serviços On-Line da Região de Trás-os-Montes e Alto Douro. Aqui encontrará disponíveis serviços de Câmaras Municipais, Centros Regionais da Segurança Social, Centros de Saúde e Direcção Regional de Agricultura. Em breve serão diponibilizados serviços de empresas privadas. Tudo de forma simples e gratuíta.
Para saber quais as entidades e os serviços que pode utilizar consulte a lista de entidades e serviços disponíveis.
Se pretende mais informações sobre o processo de adesão click em Ajuda.
Para aderir ao portal Trás-os-Montes.net click em Adesão.
Pode ainda conheçer o portal através de uma demonstração. Começará por assistir ao processo de adesão e ás instruções necessárias para a utilização dos serviços disponíveis.
Se representa uma entidade que pretende ver os seus serviços disponíveis no portal, contacte-nos e saiba como.
Condições de utilização »
Login
Nº de Utilizador: 100006001
Palavra passe: ••••••••
ok
Recuperar Palavra Passe »
Ainda não aderiu? Adira já!
Retroceder » Topo » Recomendar Site »
© 1999-2003 SCETAD - TRÁS-OS-MONTES DIGITAL. TODOS OS DIREITOS RESERVADOS.
DESENVOLVIMENTO: CI-UTAD. OPTIMIZADO PARA 800X600 IE4+
Portugal Digital
União Europeia FEDER

Figura 23: Página inicial no portal TRAS-OS-MONTES.NET

Após entrar no portal o utilizador tem a possibilidade de colocar um pedido ou consultar os processos das entidades aderentes. Para tal começa por seleccionar a opção "Serviços" sendo então apresentada a lista das entidades aderentes, como é ilustrado pela figura 24. De seguida, escolhendo uma das entidades é-lhe apresenta a página de serviços dessa entidade de acordo com a figura 25.

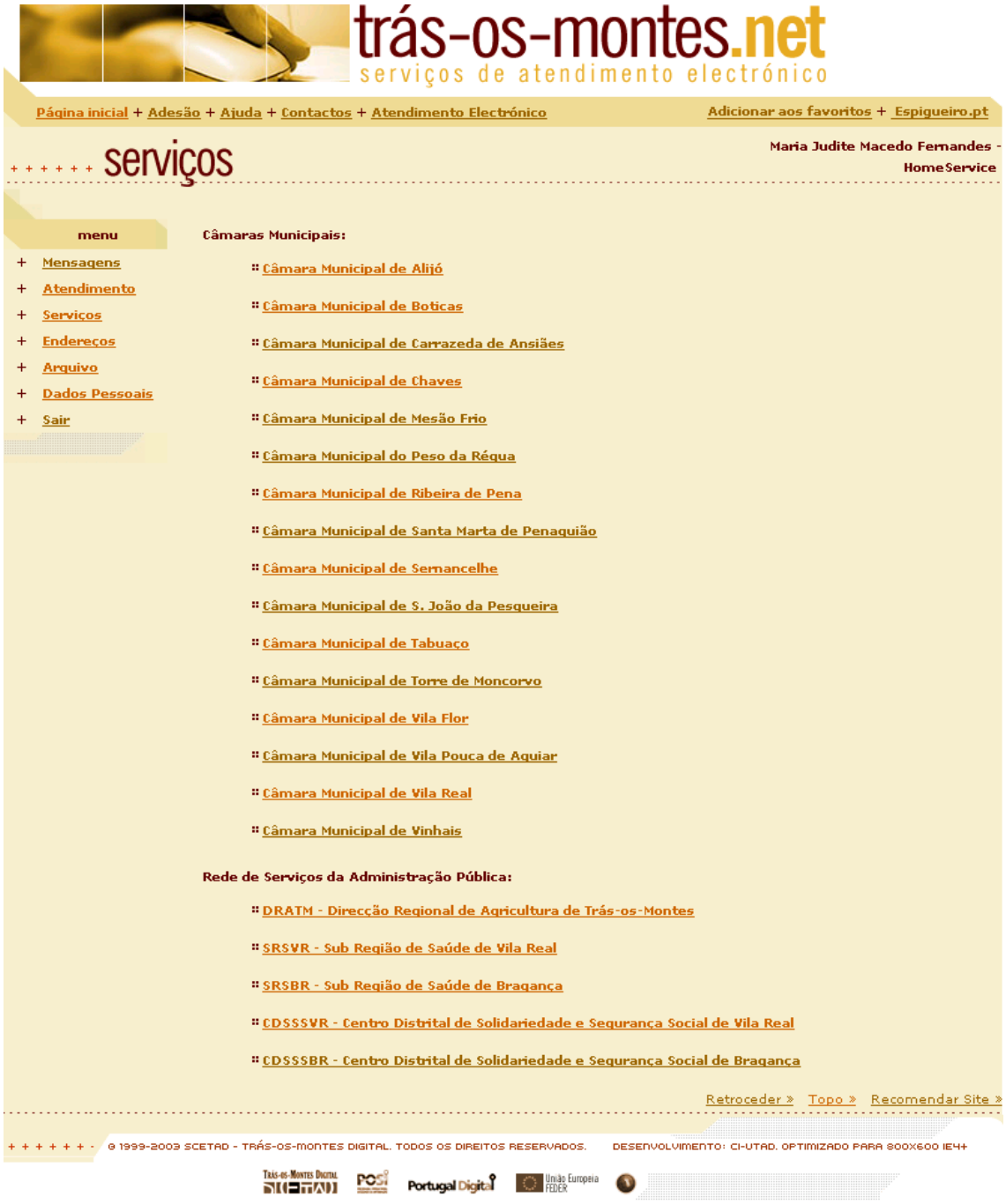
trás-os-montes.net
serviços de atendimento electrónico
Página inicial + Adesão + Ajuda + Contactos + Atendimento Electrónico
Adicionar aos favoritos + Espigueiro.pt
serviços
Maria Judite Macedo Fernandes -
HomeService
menu
+ Mensagens
+ Atendimento
+ Serviços
+ Endereços
+ Arquivo
+ Dados Pessoais
+ Sair
Câmaras Municipais:
Câmara Municipal de Alijó
Câmara Municipal de Boticas
Câmara Municipal de Carrazeda de Ansiães
Câmara Municipal de Chaves
Câmara Municipal de Mesão Frio
Câmara Municipal do Peso da Régua
Câmara Municipal de Ribeira de Pena
Câmara Municipal de Santa Marta de Penaguião
Câmara Municipal de Sernancelhe
Câmara Municipal de S. João da Pesqueira
Câmara Municipal de Tabuaço
Câmara Municipal de Torre de Moncorvo
Câmara Municipal de Vila Flor
Câmara Municipal de Vila Pouca de Aguiar
Câmara Municipal de Vila Real
Câmara Municipal de Vinhais
Rede de Serviços da Administração Pública:
DRATM - Direcção Regional de Agricultura de Trás-os-Montes
SRSVR - Sub Região de Saúde de Vila Real
SRSBR - Sub Região de Saúde de Bragança
CDSSSVR - Centro Distrital de Solidariedade e Segurança Social de Vila Real
CDSSSBR - Centro Distrital de Solidariedade e Segurança Social de Bragança
Retroceder » Topo » Recomendar Site »
© 1999-2003 SCETAD - TRÁS-OS-MONTES DIGITAL. TODOS OS DIREITOS RESERVADOS.
DESENVOLVIMENTO: CI-UTAD. OPTIMIZADO PARA 800X600 IE4+
Portugal Digital
União Europeia FEDER

Figura 24: Página de entidades aderentes no portal TRAS-OS-MONTES.NET

trás-os-montes.net
serviços de atendimento electrónico
Página inicial + Adesão + Ajuda + Contactos + Atendimento Electrónico
Adicionar aos favoritos + Espigueiro.pt
serviços disponibilizados
Maria Judite Macedo Fernandes - HomeService
menu
Mensagens
Atendimento
Serviços
Endereços
Arquivo
Dados Pessoais
Sair
Câmara Municipal de Vila Real
Serviço Disponível
Serviço Indisponível
Serviço
Disponibilidade
Alteração do mapa de horário
Apresentação de solicitações e reclamações
Pedido de limpeza de contentores cheios
Pedido de limpeza de derivados resultantes de limpezas de espaços verdes
Pedido de limpeza de lixeiras
Pedido de mapa horário
Pedido de remoção de animais mortos e abandonados
Pedido de remoção de objectos domésticos fora de uso
SMIC - Serviço municipal de informação ao consumidor
Consultar processos
Retroceder » Topo » Recomendar Site »
© 1999-2003 SCETAD - TRÁS-OS-MONTES DIGITAL. TODOS OS DIREITOS RESERVADOS.
DESENVOLVIMENTO: CI-UTAD. OPTIMIZADO PARA 800X600 IE4+

Figura 25: Página de serviços de uma entidade no portal TRAS-OS-MONTES.NET

A consulta dos processos do utilizador é possível através da página "Atendimento", tal como ilustra a figura 26.

Figura 26: Página de processos no portal TRAS-OS-MONTES.NET

A visualização do próprio documento em si é possível depois deste ser seleccionado na página "Atendimento". Um exemplo de um documento é apresentado na figura 27.

Exmo. Senhor
Presidente da Câmara Municipal de Alijó
Pedido de substituição do contador
IDENTIFICAÇÃO
Nome completo: Maria Judite Macedo Fernandes
Nº de contribuinte: 209013451
Residência em: Rua Poeta Bocage, nº 5 - 2º C
Código Postal: 1600-581
Localidade: Lisboa
País: Portugal
OBJECTO DA REQUISIÇÃO
Vem requerer a V.ª Exª se digne mandar alguém substituir o meu contador de fornecimento de água pública a que se refere a instalação Nº 123456789 , sito no lugar de Rua S. José , freguesia de Vila Verde , Concelho de Alijó, em virtude de o mesmo se encontrar avariado .
Pede deferimento, 6/20/2007
O utente: Maria Judite Macedo Fernandes
A PREENCHER PELOS SERVIÇOS
ESTADO
DESPACHO
DEFERIDO
Esta substituição será realizada no próximo dia 25/06/2007 pelas 9H00.
Serviço Cooperativo de Extensão em Trás-os-Montes e Alto Douro
Arquivar
Imprimir
Cancelar

Figura 27: Página de visualização de processo no portal TRAS-OS-MONTES.NET

Como se pode verificar, a consulta de processos no portal TRAS-OS-MONTES.NET torna transparente para o cidadão o facto de se estar a aceder ao sistema remoto WSAutarquias.

4.7 Resumo

Neste capítulo, dedicado à implementação de um SI usando Web Services, propõem-se os modelos conceptual, de arquitectura e tecnológico, de modo a obter uma solução interoperável entre os sistemas TRAS-OS-MONTES.NET e WSAutarquias.

Com a introdução de WS criou-se uma camada de abstração, permitindo disponibilizar aplicações e informação como serviços e assim criar pontos de integração normalizados. Esta arquitectura segue a filosofia de *Service Oriented Arquitecture* (SOA) e permite a reutilização e integração destes serviços em soluções mais abrangentes e flexíveis.

O protótipo de integração e cada dos seus componentes foram apresentados. Durante o desenvolvimento do protótipo os modelos e protocolos abordados não foram esquecidos, no entanto, as plataformas de desenvolvimento escolhidas tornam transparente todo o esforço de criação e interacção com as mensagens SOAP, bem como a criação e utilização dos documentos WSDL que descrevem os WS, facilitando assim a sua programação.

Demonstrou-se, com a criação do protótipo funcional, que a interoperabilidade entre SI usando WS é efectiva, permitindo, desta forma, que cidadãos possam interagir com serviços de diversas entidades numa única plataforma. Permitiu também tirar partido das possibilidades que a *Internet* tem de se transformar numa plataforma de disponibilização de serviços.

C A P Í T U L O V

5. Conclusões

Neste capítulo encerra-se esta dissertação, acreditando que os objectivos inicialmente propostos foram alcançados. De seguida faz-se uma síntese final dos principais temas apresentados ao longo dos capítulos anteriores e tecem-se alguns comentários, que se julgam oportunos para a conclusão deste trabalho.

Esta dissertação teve como objectivo estudar e experimentar as tecnologias de integração e testar num projecto piloto a plataforma desenvolvida com base no modelo de arquitectura de Web Services. Para tal, optou-se por começar com a apresentação das principais características, conceitos e tecnologias dos sistemas distribuídos, para de seguida se abordar de forma mais exaustiva os WS e os seus protocolos associados.

O trabalho realizado permitiu verificar que os WS se adaptam bem a situações em que é necessário o acesso a componentes de software remotos de uma maneira simples e em que haja suporte em várias plataformas.

O problema estudado neste trabalho pertence à classe mais complexa dos problemas de SI distribuídos para integração e interoperabilidade de aplicações utilizando protocolos da *Internet*. A sua importância é bem visível quando as organizações precisam que as suas aplicações comuniquem entre si, quer seja internamente quer seja com outras aplicações externas.

Ao longo desta tese de mestrado, pretendeu-se dar uma contribuição para a investigação dos WS, um tema emergente e recheado de potencialidades. No entanto, o tema dos WS não fica esgotado de forma nenhuma. Este assunto é actual e como tal

encontra-se em constante evolução. Assim, apresentam-se de seguida algumas considerações.

A utilização de WS pode ser classificada para as duas grandes áreas: *Intranet* e *Internet*, sendo que os dois casos envolvem o mesmo conjunto de tecnologias, mas com uma diferença na sua utilização, que é determinada pelos requisitos de segurança que terão de ser garantidos.

Na *Intranet* os WS são expostos somente dentro de uma zona protegida por uma *firewall*, o facto de serem baseados em protcolos-padrão deverá diminuir o custo de integração entre aplicações e terão preferência sobre outras soluções. No entanto, o maior interesse sobre a adopção de WS é a sua utilização na *Internet*, viabilizada por factores básicos de performance, escalabilidade e segurança, prometendo tornar a *Internet* num grande manancial de serviços.

Um dos principais desafios que actualmente se coloca às soluções de integração baseadas em WS é o tema da segurança, o qual foi superficialmente referido nesta dissertação. Os desafios prendem-se essencialmente com a autenticação e autorização no momento da utilização dos serviços.

Existem padrões estabelecidos para segurança ao nível de transporte (SSL) e segurança de mensagens (SMIME). Mas, ainda não foi gerado um consenso, sobretudo pela enorme complexidade e exigência subjacente aos requisitos, além disso, a tecnologia de suporte ainda não atingiu um grau de maturidade e de abrangência necessário.

A emergência dos WS vem adicionar alguma complexidade à gestão das redes de comunicação. Perante o potencial de adopção dos WS, é bastante provável que o fluxo de comunicações venha a crescer exponencialmente. Este aumento de mensagens XML na rede vai implicar um congestionamento das redes de comunicação, pelo que terá de se optar por um aumento da largura de banda disponível, sob pena de que a performance e os níveis de serviço venham a pôr em causa os ritmos de disseminação dos WS.

Apesar da capacidade e facilidade da utilização dos WS na integração de aplicações, algumas questões fundamentais ainda não têm uma solução bem-estruturada e inibem a adopção desta tecnologia com maior frequência. Um dos inibidores mais significativos é a dificuldade em definir e manter o contexto de uma transacção, que pode incluir um conjunto de actividades críticas. Caso alguma dessas acções não seja possível, toda a transação deveria poder ser inviabilizada.

Os WS carecem actualmente de um estado de maturação que permita a sua adopção em massa pela comunidade informática. O constante aparecimento de novas versões e especificações não permite um correcto amadurecimento dos conceitos envolvidos, impedindo assim que a sociedade informática tenha tempo para as assimilar e colocar em prática.

A utilização de WS numa primeira fase, somente nas *Intranets* (figura 28), permitiu uma melhoria das funcionalidades e da qualidade da interacção com as aplicações, possibilitando às organizações a necessária adaptação à tecnologia e o retorno mais rápido do investimento, sem necessidade de grandes preocupações com a segurança.

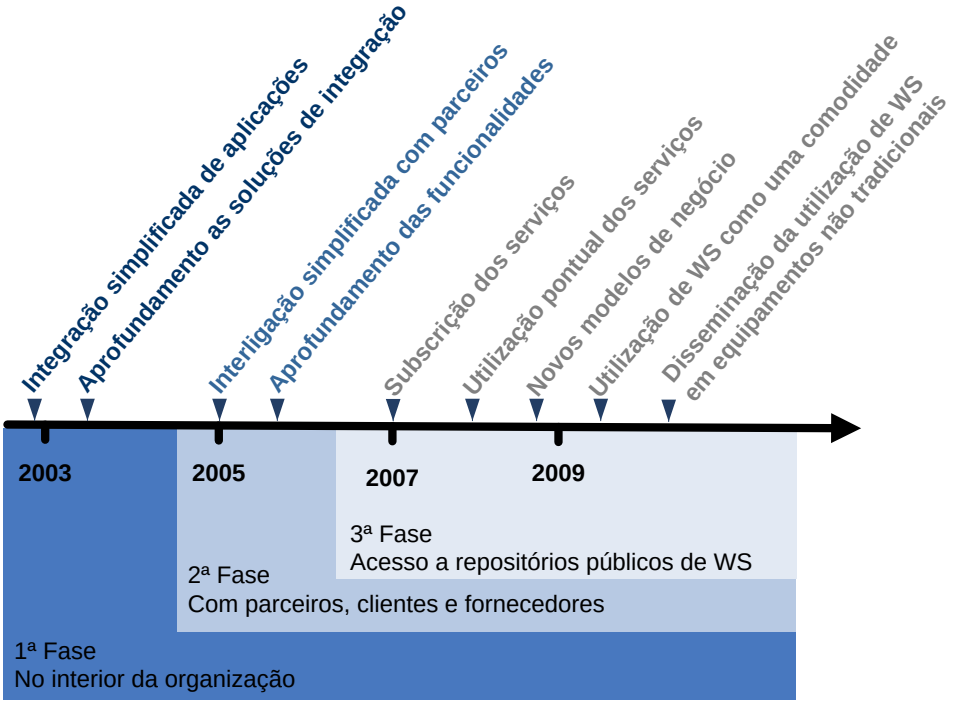

Figura 28: Fases da adopção dos Web Services [IDC 2002]

Numa segunda fase, as organizações avançam para a integração dos seus sistemas com os dos seus parceiros. Nesta fase, que é suportada por uma maior maturidade da tecnologia, os aspectos de segurança são fulcrais para a disseminação dos WS.

A terceira, envolverá o reconhecimento da necessidade de dotar as aplicações com determinadas funcionalidades. Estas serão procuradas em repositórios de acesso público, de modo a encontrar os WS mais apropriados.

Além disso, após a determinação dos WS mais convenientes, as aplicações serão também capazes de os utilizarem dinâmicamente, sem qualquer tipo de intervenção humana. Será nesta fase que a exigência sobre os requisitos da infra-estrutura de WS se tornará mais forte, o que leva a crer que as melhorias suplementares só serão implementadas, após um período de plena adaptação aos WS.

Por forma a preparar e a adaptar os produtos à disseminação dos WS, a indústria de software tem vindo a envolver-se num esforço intenso de desenvolvimento tecnológico. Algumas das empresas mais bem sucedidas e influentes de desenvolvimento de TI, não só adoptam tecnologias de interoperabilidade como também desenvolvem as suas próprias especificações adicionais para fazer com que a interoperabilidade se estenda ao domínio de transacções confiáveis.

O sucesso dos WS passa necessariamente pela existência de sinergias alargadas entre a generalidade da indústria, pela partilha e abertura dos processos de normalização e das próprias especificações daí resultantes. Efectivamente, as empresas têm sido exemplares na sua generalidade a respeitar a compatibilidade.

Esta universalidade já havia sido atingida com as páginas da Web. Todos os sites se comportam mais ou menos da mesma maneira e têm mais ou menos a mesma aparência, independentemente do tipo de computador ou browser utilizado para acedê-lo. Os protocolos da Web revolucionaram a forma como as pessoas se relacionam com a tecnologia.

As tecnologias subjacentes aos web services, como XML, SOAP, UDDI e WSDL, são abertas, amplamente divulgadas e consensuais. Por outro lado, potenciam

uma real independência das linguagens de programação, das arquitecturas de computadores e sistemas operativos, o que permite uma evolução mais suave e económica para este paradigma computacional.

Dos resultados alcançados com este trabalho, salienta-se que este caso de estudo serviu para discussão dos aspectos ligados à integração e inteoperabilidade entre SI. Em particular, o desenvolvimento de um protótipo utilizando WS, permitiu o aprofundamento dos aspectos relacionados com as suas tecnologias subjacentes.

Como trabalho futuro, propõe-se, com base nos desenvolvimentos realizados neste trabalho, evoluir o portal TRÁS-OS-MONTES.NET por forma a que este se integre com serviços de outras entidades. Por um lado, existem as entidades que já disponibilizam serviços no portal, mas estes não estão integrados com os seus SI, como é o caso de Direcções Regionais de Agricultura, Sub-regiões de Sáude, Centros Distritais de Solidariedade e Segurança Social, por outro lado, procurar novas entidades para aderir ao projecto.

Finalmente, sublinha-se que a capacidade dos Web Services para a integração rápida e eficaz de aplicações, dados e funcionalidades, é fundamental para a concretização dos objectivos estratégicos das organizações, como é o caso do protótipo apresentado nesta dissertação.

R E F E R Ê N C I A S

[Amaral e Varajão 2000] Amaral, A. M. e J. E. Q. Varajão, *Planeamento de Sistemas de Informação*, FCA, 2000.

[Bulas Cruz e Ramos 2001] Bulas Cruz, J. A. e L. Ramos, "O Serviço Cooperativo de Extensão em Trás-os-Montes e Alto Douro (SCETAD), uma experiência pioneira de utilização das NTIC na modernização dos serviços públicos locais em meio rural", http://www.arribes.net/fuenoticia/congreso-arribes1/ponencias, 2001.

[Boar 2003] Boar, C., *XML Web Services in the Organization*, Microsoft Corporation, Washington, 2003.

[Booth *et al.* 2004] Boot, D., H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris e D. Orchard, "Web Services Architecture", World Wide Web Consortium Working Group, http://www.w3.org/TR/2004/NOTE-ws-arch/, 2004.

[Buxton 1980] Buxton, J., *Requirements for Ada programming support environments: stoneman*, Washington, 1980.

[CBDi 1999] CBDi Forum, "Component-Based Development", Forum for Componet Based Development and Integration Report, http://www.cbdiforum.com, 1999.

[Chappell 2003] Chappell, D., "New Technologies Help You Make Your Web Services More Secure", http://www.davidchappell.com/articles, 2003.

[Christensen *et al.* 2001] Christensen, E., F. Curbera, G. Meredith e S. Weerawarana, "Web Services Description Language (WSDL) 1.1", World Wide Web Consortium, http://www.w3.org/TR/wsdl, 2001.

[Coulouris *et al.* 1994] Coulouris, G., J. Dollimore e T. Kindberg, *Distributed Systems: Concepts and Design*, 3th edition, Addison-Wesley, 2000.

[Espigueiro 2002] Espigueiro, Portal de Informações Regionais, http://www.espigueiro.pt/servico_cooperativo/index.html, 2002.

[Gardner 2003] Gardner, Dana, "Interoperability emerges as new core competency for enterprise architects", Yankee Group, http://yankeegroup.com, 2003.

[Gordon 2000] Gordon, A., *The COM and COM+ Programing*, Prentice-Hall, 2000.

[Gudgin *et al.* 2003] Gudgin, M., M. Hadley, N. Mendelsohn, j. Moreau e H. F. Nielsen, "SOAP Version 1.2 Part 1: Messaging Framework", World Wide Web Consortium, http://www.w3.org/TR/soap12-part1/, 2003.

[Heiler 1995] Heiler, S., *Semantic Interoperability*, 2nd edition, vol. 27, ACM Computing Surveys, 1995.

[Hirschfeld 1996] Hirschfeld, R., "Three-Tier Distribution Architecture", Plop'96, Ilinois, Washington University, 1996.

[IDC 2002] IDC Portugal, "Web Services", Caderno IDC, 2002.

[Iwasa 2004] Iwasa K., "Web Services Reliable Messaging TC WS-Reliability 1.1", http://docs.oasis-open.org/wsrm/ws-reliability/v1.1, 2004.

[IBM 2000] IBM Web Services Architecture Team, "Web Services architecture overview, the next stage of evolution for e-business", http://www.ibm.com/developerworks/webservices/library/w-ovr/?dwzone=webservices, 2000.

[IETF 2007] Internet Engineering Task Force, http://www.ietf.org, 2007.

[Shortcut 2003] Shortcut, "Manual de Utilização – WsAutarquias SHSecurityClient v1.0 para a Medidata", 2003.

[Shortcut 2007] Shortcut, http://www.shortcut.pt, 2007.

[Smith 2007] Smith, J., *Windows Comunication Foundation*, Microsoft Press, Washington , 2007.

[Lopes e Ramalho 2005] Lopes, C. J. F. e J. C. Ramalho, *Web Services: Aplicações Distribuídas sobre Protocolos Internet*, FCA, 2005.

[Manola 1995] Manola, F., *Interoperability Issue in Large-Scale Distributed Object System*, 2nd edition, vol. 27, ACM Computing Surveys, 1995.

[Matos 2001] Matos, L. M. C., "Integração de Sistemas de manufactura", *Revista INGENIUM - Ordem dos Engenheiros*, 2ª série, nº 56, 2001, pp.68-74.

[Medidata 2007] Medidata, http://www.medidata.pt, 2007.

[Microsoft 2001a] Microsoft Corporation, "Course Developing XML Web Services Using ASP.NET", 2001.

[Microsoft 2001b] Microsoft Corporation, "Release History for Visual Basic", Microsoft Corporation, http://msdn.microsoft.com/vbasic/bday/history.asp, 2001.

[MSDN 2007] MSDN, "SOAP Message Modification Using SOAP Extensions", http://msdn2.microsoft.com/en-us/library/esw638yk.aspx, 2007.

[Navarro et al. 2000] Navarro, A.; C. White e L. Burman, *Mastering XML*, Sybex, USA, 2000.

[OASIS 2006] Organization for the Advancement of Structured Information Standards, "Web Services Security SOAP Messages with Attachments (SwA) Profile 1.1" http://docs.oasis-open.org/wss/v1.1, 2006.

[OASIS 2007a] Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org, 2007.

[OASIS 2007b] Organization for the Advancement of Structured Information Standards, "Security Assertion Markup Language (SAML)", http://xml.coverpages.org/saml.html, 2007.

[OMG 2001] OMG, "The Common Object Request Broker: Architecture and Specification", Object Management Group, http://www.omg.org, revision 2.6, USA, 2001.

[OMG 2004]	OMG, "Java, RMI and CORBA", http://www.omg.org/library/wpjava.html, 2004.
[Orfali e Harkey 1998]	Orfali, R. e D. Harkey, *Client/Server Programming with Java and CORBA*, 2nd edition, John Wiley & Sons, New York, 1998.
[POSI 2003]	POSI, Programa Operacional Sociedade da Informação, http://www.posi.pcm.gov.pt, 2003.
[Promon 2002]	Promon Tecnologia, "O Modelo de Web Services - Como Desenvolver Aplicações numa nova Arq u i t e t u ra", Business & Technology Review Series, 2002.
[Ramalho e Henriques 2002]	Ramalho, J. e P. Henriques, *XML & XSL Da Teoria à prática*, FCA, 2002.
[Roque 2003]	Roque, V. e J. Oliveira, *CORBA, DCOM e JavaRMI – Uma Análise Comparativa*, IPG / IETA, 2003.
[Seetharaman 1998]	Seetharaman, K., *The CORBA Connection*, 10th edition, vol. 41, Comunication of the ACM, 1998.
[Short 2002]	Short, S., Building XML Web Services for the Microsoft .Net Platform, Microsoft Corporation, Washington, 2002.
[Silva 2003]	Silva, M. M., *Integração de Sistemas de Informação*, FCA, 2003.
[SOAP 2003]	SOAPBuilders Interoperability Lab, http://www.xmethods.net/ilab, 2003.
[Sommerville 2001]	Sommerville, I., *Software Engineering*, 6 th edition, Pearson Education Limited, 2001.
[Snell *et al.* 2001]	Snell, J., D. Tidwell e P. Kulchenko, Programming Web Services with SOAP, O'Reilly, 2001.
[Sun 1999]	SUN, "JINI Technology Architectural Overview", http://www.sun.com/jini/whitepapers/architecture.html, 1999.
[Tanenbaum 2002]	Tanenbaum, A. S., *Distributed Systems: Principles and Paradigms*, Prentice-Hall, 2002.

[Trowbridge 2004] Trowbridge, D., U. Roxburgh, G. Hohpe, D. Manolescu e E. G. Nadhan, Veríssimo, P e L. Rodrigues, "Integration Patterns", Microsoft Patterns & Practices, 2004.

[UTAD 2004] Universidade de Trás-os-Montes e Alto Douro e Microsoft Corporation, "Portal Transaccional aproxima a Sociedade de Informação deste canto remoto da Europa", http://download.microsoft.com/download/8/6/5/8650b90b-a5a6-42cc-bb1a-f4a50decbe53/TrasosMontes.doc, 2004.

[W3C 2002a] World Wide Web Consortium, "XML Signature Syntax and Processing", http://www.w3.org/tr/xmldsig-core, 2002.

[W3C 2002b] World Wide Web Consortium, "XML Encryption Syntax and Processing", http://www.w3.org/tr/xmlenc-core, 2002.

[W3C 2004] World Wide Web Consortium, "Web Services Architecture Working Group", http://www.w3.org/tr/ws-arch, 2004.

[W3C 2005] World Wide Web Consortium, "Naming and Addressing: URIs, URLs, …", http://www.w3.org/Addressing, 2006.

[W3C 2006a] World Wide Web Consortium, "Web Services Addressing 1.0 - Core", http://www.w3.org/tr/ws-addr-core, 2006.

[W3C 2006b] World Wide Web Consortium, "Extensible Markup Language", http://www.w3.org/tr/rec-xml, 2006.

[W3C 2006c] World Wide Web Consortium, "MTOM Serialization Policy Assertion", http://www.w3.org/Submission/WS-MTOMPolicy, 2006.

[W3C 2007a] World Wide Web Consortium, "Web Services Description Language Version 2.0", http://www.w3.org/tr/wsdl20, 2007.

[W3C 2007b] World Wide Web Consortium, "XML Schema 1.1", http://www.w3.org/xml/Schema, 2007.

[W3C 2007c] World Wide Web Consortium, "Semantic Annotations for WSDL and XML Schema", http://www.w3.org/tr/sawsdl, 2007.

[W3C 2007d] World Wide Web Consortium, "SOAP Version 1.2 Part 1: Messaging Framework", http://www.w3.org/tr/SOAP12-part1, 2007.

[W3C 2007e] World Wide Web Consortium, http://www.w3.org, 2007.

[W3Schools 2007] W3Schools, “XML Schema Tutorial”, http://www.w3schools.com/schema/default.asp, 2007.

[WS-I 2005] Web Services Interoperability Organization, “Interoperability: Ensuring the Success of Web Services - an overview to WS-I”, http://www.ws-i.org/about/Default.aspx, 2005.

[WS-I 2006] Web Services Interoperability Organization, “Working Group Charter: Reliable Secure Profile 1.0”, http://www.ws-i.org/docs/charters/RSP_Charter1-0.pdf, 2006.

[WS-I 2007a] Web Services Interoperability Organization, http://www.ws-i.org, 2007.

[WS-I 2007b] Web Services Interoperability Organization, “Working Group Charter: Basic Profile 1.2 and 2.0”, http://www.ws-i.org/Profiles/BasicProfile-1.2.html, 2007.

[WS-I 2007c] Web Services Interoperability Organization, “Basic Security Profile Version 1.0”, http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html, 2007.

[Wegner 1996] Wegner, P., *Interoperability*, vol. 28, ACM Computing Surveys, 1996.

Anexo I

Descrição WSDL dos Web Services do sistema WSAutarquias

Este anexo apresenta a descrição WSDL dos Web Services do sistema WSAutarquias.

Todos os *schemas* de retorno devolvem no objecto *dataset* a tabela Erros, se o número de retorno for igual ou maior do que 1, então houve sucesso, se for negativo, ocorreu um erro, de acordo com o Web Service correspondente.

```
<xs:element name="Erros">
 <xs:complexType>
   <xs:sequence>
    <xs:element name="Numero" type="xs:string" minOccurs="0" />
    <xs:element name="Descricao" type="xs:string" minOccurs="0" />
   </xs:sequence>
  </xs:complexType>
</xs:element>
```

1. Web Service Autenticação

O URL para chamada a este Web Service é o seguinte, onde `ip_servidor_web` é o endereço IP onde foi instalado o wsAutarquias.
http://ip_servidor_web/WSACentral/wsAAuthentication.asmx.

Este Web Service é responsável por solicitar ao servidor uma chave para encriptação e ainda para solicitar uma credencial autenticada, através dos seguintes Web Methods:

1.1. GetValidKey
1.2. Authenticate

WSDL do Web Service:

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="http://ws.shortcut.pt/ws.WSAutarquiasCentral/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://ws.shortcut.pt/ws.WSAutarquiasCentral/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
  <s:schema elementFormDefault="qualified"
targetNamespace=http://ws.shortcut.pt/ws.WSAutarquiasCentral/>
    <s:element name="getValidKey">
      <s:complexType />
    </s:element>
    <s:element name="getValidKeyResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="getValidKeyResult"
type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="Authenticate">
      <s:complexType>
```

```
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="client" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="pass" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="AuthenticateResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="AuthenticateResult"
type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SHWSClientCredentials" type="s0:SHWSClientCredentials" />
    <s:complexType name="SHWSClientCredentials">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="SHWSCredentialId" type="s:string"/>
      </s:sequence>
    </s:complexType>
    <s:element name="teste">
      <s:complexType />
    </s:element>
    <s:element name="testeResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="testeResult" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
</types>
<message name="getValidKeySoapIn">
  <part name="parameters" element="s0:getValidKey" />
</message>
<message name="getValidKeySoapOut">
  <part name="parameters" element="s0:getValidKeyResponse" />
</message>
<message name="AuthenticateSoapIn">
  <part name="parameters" element="s0:Authenticate" />
</message>
<message name="AuthenticateSoapOut">
  <part name="parameters" element="s0:AuthenticateResponse" />
</message>
<message name="AuthenticateSHWSClientCredentials">
  <part name="SHWSClientCredentials" element="s0:SHWSClientCredentials" />
</message>
<message name="testeSoapIn">
  <part name="parameters" element="s0:teste" />
</message>
  <message name="testeSoapOut">
  <part name="parameters" element="s0:testeResponse" />
</message>
<message name="testeSHWSClientCredentials">
  <part name="SHWSClientCredentials" element="s0:SHWSClientCredentials" />
</message>
<portType name="WSAAuthenticationSoap">
  <operation name="getValidKey">
    <input message="s0:getValidKeySoapIn" />
    <output message="s0:getValidKeySoapOut" />
  </operation>
  <operation name="Authenticate">
    <input message="s0:AuthenticateSoapIn" />
    <output message="s0:AuthenticateSoapOut" />
  </operation>
  <operation name="teste">
    <input message="s0:testeSoapIn" />
    <output message="s0:testeSoapOut" />
  </operation>
</portType>
<binding name="WSAAuthenticationSoap" type="s0:WSAAuthenticationSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="getValidKey">
```

```
    <soap:operation
soapAction="http://ws.shortcut.pt/ws.WSAutarquiasCentral/getValidKey" style="document"/>
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="Authenticate">
    <soap:operation
soapAction="http://ws.shortcut.pt/ws.WSAutarquiasCentral/Authenticate"
style="document"/>
    <input>
      <soap:body use="literal" />
      <soap:header message="s0:AuthenticateSHWSClientCredentials"
part="SHWSClientCredentials" use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="teste">
    <soap:operation soapAction="http://ws.shortcut.pt/ws.WSAutarquiasCentral/teste"
style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="s0:testeSHWSClientCredentials" part="SHWSClientCredentials"
use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
<service name="WSAAuthentication">
  <documentation>WSAAuthentication no Host Central</documentation>
  <port name="WSAAuthenticationSoap" binding="s0:WSAAuthenticationSoap">
    <soap:address location="http://ws.shortcut.pt/WSACentral/wsAAuthentication.asmx" />
  </port>
</service>
</definitions>
```

2. Web Service Munícipe

O URL para chamada a este Web Service é:

http://ip_servidor_web/WSACentral/wsMunicipe.asmx.

Este Web Service define os seguintes Web Methods.

2.1. Autentica

Método Responsável pela autenticação de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <Autentica xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <PIN>string</PIN>
    </Autentica>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe (cifrado com a credencial em uso)
Password – VARCHAR(4) – PIN do Munícipe (cifrado com a credencial em uso)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="Autentica">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="NOME" type="xs:string" minOccurs="0" />
                <xs:element name="ACESSOS" type="xs:int" minOccurs="0" />
                <xs:element name="ACESSOSTOTAIS" type="xs:int" minOccurs="0" />
                <xs:element name="ULTIMOACESSO" type="xs:dateTime" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
```

```
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
NOME – Nome do Munícipe
ACESSOS – Número. de acessos do Munícipe desde a última alteração de PIN efectuada pelo sistema (Nova adesão de Munícipe ou Geração automática do PIN pelo AnswerDesafio). Caso seja 1, deve ser sugerida, ao Munícipe, a alteração do PIN.
ACESSOSTOTAIS – Total de acessos do Munícipe
ULTIMOACESSO – Data / Hora do último acesso efectuado pelo Munícipe

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Password Incorrecta. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-205 – Ultrapassou o número de tentativas erradas. A conta está temporariamente bloqueada. Neste caso, o único acesso passa a ser pelo Método Desafio.

2.2. GetMunicipe

Retorna um dataset com os dados de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetMunicipe xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
    </GetMunicipe>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
```

```
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetMunicipe">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="NOME" type="xs:string" minOccurs="0" />
                <xs:element name="MORADA" type="xs:string" minOccurs="0" />
                <xs:element name="LOCAL" type="xs:string" minOccurs="0" />
                <xs:element name="ESTADO_CIVIL" type="xs:string" minOccurs="0" />
                <xs:element name="FILIACAO_PAI" type="xs:string" minOccurs="0" />
                <xs:element name="FILIACAO_MAE" type="xs:string" minOccurs="0" />
                <xs:element name="PROFISSAO" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_NASC" type="xs:dateTime" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="BI" type="xs:string" minOccurs="0" />
                <xs:element name="BI_DATA" type="xs:dateTime" minOccurs="0" />
                <xs:element name="BI_ARQ" type="xs:string" minOccurs="0" />
                <xs:element name="PASSWORD" type="xs:string" minOccurs="0" />
                <xs:element name="HORA_CARR" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOGINS" type="xs:int" minOccurs="0" />
                <xs:element name="FREG_RESID" type="xs:string" minOccurs="0" />
                <xs:element name="FREG_RESID_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_RESID" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_RESID_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_RESID" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_RESID_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="FREG_NATUR" type="xs:string" minOccurs="0" />
                <xs:element name="FREG_NATUR_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_NATUR" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_NATUR_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_NATUR" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_NATUR_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="CODIGO_POSTAL" type="xs:string" minOccurs="0" />
                <xs:element name="TELEFONE" type="xs:string" minOccurs="0" />
                <xs:element name="FAX" type="xs:string" minOccurs="0" />
                <xs:element name="EMAIL" type="xs:string" minOccurs="0" />
                <xs:element name="SEXO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPO_USER" type="xs:short" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
NOME – Nome do Munícipe
MORADA – Morada do Munícipe
LOCAL – Localidade
ESTADO_CIVIL – Estado Civil
FILIACAO_PAI – Nome do Pai
FILIACAO_MAE – Nome da Mãe
PROFISSAO – Profissão do Munícipe
DATA_NASC – Data de Nascimento do Munícipe
AUTARQUIA – Código da Autarquia
BI – Número do Bilhete de Identidade
BI_DATA – Data de Emissão do Bilhete de Identidade
BI_ARQ – Arquivo de Emissão do Bilhete de Identidade
PASSWORD – PIN Cifrado
HORA_CARR – Data / Hora em que o Munícipe se Autenticou
LOGINS – Campo mantido para retro-compatibilidade que deverá desaparecer.
FREG_RESID – Código da Freguesia de Residência
FREG_RESID_DESC –Freguesia de Residência

CONC_RESID – Código do Concelho de Residência
CONC_RESID_DESC – Concelho de Residência
DISTR_RESID – Código do Distrito de Residência
DISTR_RESID_DESC – Distrito de Residência
FREG_NATUR – Código da Freguesia da sua Naturalidade
FREG_NATUR_DESC – Freguesia da sua Naturalidade
CONC_NATUR – Código do Concelho da sua Naturalidade
CONC_NATUR_DESC – Concelho da sua Naturalidade
DISTR_NATUR – Código do Distrito da sua Naturalidade
DISTR_NATUR_DESC – Distrito da sua Naturalidade
CODIGO_POSTAL – Código Postal de Residência
TELEFONE – Telefone
FAX – Fax
EMAIL – Email
SEXO – Sexo do Munícipe (F – Feminino / M – Masculino / NULL ou N – Não especificado)
TIPO_USER – Tipo de Utilizador (0 – Utilizador Individual / 1 – Utilizador Empresarial / 2 – Funcionário)

Códigos de Erro:

1 – Sucesso
-100 – Não existe nenhum registo na Autarquia
-101 – Existem dois ou mais registos na Autarquia
-102 – Autarquia não Disponível
-104 – Autarquia em Manutenção
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente

2.3. SetDesafio

Permite definir um desafio que poderá ser usado para gerar uma nova password.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <SetDesafio xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <PIN>string</PIN>
      <Pergunta>string</Pergunta>
      <Resposta>string</Resposta>
    </SetDesafio>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)

PIN – VARCHAR(4) – PIN (cifrado com a credencial em uso)
Pergunta – VARCHAR(200) – Pergunda Desafio (cifrado com a credencial em uso)
Resposta – VARCHAR(200) – Resposta ao Desafio (cifrado com a credencial em uso)

Retorno do DataSet com Códigos de Erro.

Códigos de Erro:

1 – Sucesso
-200 – Munícipe Inexistente
-204 – Password Incorrecta. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-205 – Ultrapassou o número de tentativas erradas. A conta está temporariamente bloqueada. Neste caso, o único acesso passa a ser pelo Método Desafio.

2.4. AskDesafio

Devolve a pergunta Desafio.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <AskDesafio xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <num_contribuinte>string</num_contribuinte>
    </AskDesafio>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe (cifrado com a credencial em uso)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="AskDesafio">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
```

```
              <xs:element name="PERGUNTA" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
 </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
PERGUNTA – Pergunta Desafio

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Acesso negado. Neste caso, o único acesso passa a ser pelo Método Desafio.
-206 – Pergunta Desafio Inexistente.

2.5. AnswearDesafio

Resposta à pergunta Desafio. Em caso de sucesso gera um novo PIN automaticamente e retorna-o.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <AnswerDesafio xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <num_contribuinte>string</num_contribuinte>
      <Resposta>string</Resposta>
    </AnswerDesafio>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe (cifrado com a credencial em uso)
Resposta – VARCHAR(200) – Resposta ao Desafio (cifrado com a credencial em uso)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Læcale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="AnswerDesafio">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="PIN" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
PIN – Novo PIN gerado pelo Sistema

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Acesso negado.
-207 – Resposta ao desafio errada. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-208 – Ultrapassou o número de tentativas erradas para o Desafio. A conta está temporariamente bloqueada. Neste caso, o utilizador tem mesmo que aguardar as X horas para tentar novamente.

2.6. ChangePassword

Método que permite a alteração do PIN do munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance mlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope /">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <ChangePassword xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <PINAntigo>string</PINAntigo>
      <PINNovo>string</PINNovo>
    </ChangePassword>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)
PasswordAntiga – VARCHAR(4) – PIN actualmente utilizado (cifrado com a credencial em uso)
PasswordNovo – VARCHAR(4) – Novo PIN (cifrado com a credencial em uso)

Retorno de DataSet com Códigos de Erro.

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Password Incorrecta. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-205 – Ultrapassou o número de tentativas erradas. A conta está temporariamente bloqueada. Neste caso, o único acesso passa a ser pelo Método Desafio.

2.7. LimparMunicipe

Limpa toda a informação relacionada com o Munícipe da cache local (ex. Processo, Perfil, Requerimentos, etc). Este método deverá sempre ser chamado quando um munícipe abandona a sessão web.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparMunicipe xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparMunicipe>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de contribuinte

Retorno de DataSet com Códigos de Erro.

Códigos de Erro:

1 – Sucesso
-200 – Munícipe Inexistente

3. Web Service Processos

O URL que define esta classe de web services é:

http://ip_servidor_web/WSACentral/wsProcessos.asmx.

3.1. LimparProcessos

Elimina todos os processos de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparProcessos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparProcessos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

3.2. GetProcessos

Retorna um dataset com os processos de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetProcessos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
```

```
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
    </GetProcessos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetProcessos">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC_DESCR" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ABERT" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOCAL_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
                <xs:element name="PROPRIETARIO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC_DESCR – Descrição do Tipo de Processo
DATA_ABERT – Data de Abertura do Processo
LOCAL_OBRA – Descrição do Local da Obra
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
PROPRIETARIO – Nome do Proprietário do Processo
TIPOPROC – Código do Tipo de Processo
DESCRICAO – Descrição do Processo
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra

RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-201 – Munícipe sem Processos.

Ordenação dos Resultados: IDPERFIL ASC, DATA_ABERT DESC

3.3. GetProcesso

Retorna um dataset com os dados do processo indicado.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetProcesso xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetProcesso>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
```

```
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetProcesso">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC_DESCR" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ABERT" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOCAL_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
                <xs:element name="PROPRIETARIO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC_DESCR – Descrição do Tipo de Processo
DATA_ABERT – Data de Abertura do Processo
LOCAL_OBRA – Descrição do Local da Obra
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
PROPRIETARIO – Nome do Proprietário do Processo
TIPOPROC – Código do Tipo de Processo
DESCRICAO – Descrição do Processo
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.

3.4. PesqProcessos

Pesquisa determinado texto em Processos.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <PesqProcessos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
    </PesqProcessos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="PesqProcessos">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC_DESCR" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ABERT" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOCAL_OBRA - Descrição do Local da Obra"
type="xs:string" minOccurs="0" />
                <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
                <xs:element name="PROPRIETARIO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA - Número de Contribuinte do Responsável pela
Responsável pela Direcção Técnica da Obra" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC_DESCR – Descrição do Tipo de Processo
DATA_ABERT – Data de Abertura do Processo
LOCAL_OBRA – Descrição do Local da Obra
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
PROPRIETARIO – Nome do Proprietário do Processo
TIPOPROC – Código do Tipo de Processo
DESCRICAO – Descrição do Processo
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-304 – Não foram encontrados Processos com o texto a pesquisar

Ordenação dos Resultados: IDPERFIL ASC, DATA_ABERT DESC

4. Web Service Requerimentos

O URL que define esta classe de web services é http://ip_servidor_web/WSACentral/wsRequerimentos.asmx.

4.1. LimparRequerimentos

Elimina todos os requerimentos de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparRequerimentos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparRequerimentos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

4.2. LimparRequerimentosProcesso

Elimina todos os requerimentos de processo, de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparRequerimentosProcesso xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
```

```
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </LimparRequerimentosProcesso>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente
-300 – Processo inexistente

4.3. GetRequerimentos

Retorna um dataset com os requerimentos de um processo, de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetRequerimentos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetRequerimentos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
```

```
        <xs:element name="GetRequerimentos">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
              <xs:element name="REQUER" type="xs:string" minOccurs="0" />
              <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_ENTRADA" type="xs:dateTime" minOccurs="0" />
              <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
              <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
              <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
              <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
              <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
              <xs:element name="IDPERFIL" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
REQUER – Número do Requerimento
DESCRICAO – Descrição do Requerimento
DATA_ENTRADA – Data de Entrada do Requerimento
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao processo.
-300 – Processo inexistente.
-301 – Processo sem requerimentos

Ordenação dos Resultados: IDPERFIL ASC, DATA_ENTRADA DESC

4.4. GetRequerimento

Retorna um dataset com os dados do requerimento indicado.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetRequerimento xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
      <IDrequerimento>string</IDrequerimento>
    </GetRequerimento>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo
IDRequer – VARCHAR(15) - Número do Requerimento

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="GetRequerimento">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
               <xs:element name="REQUER" type="xs:string" minOccurs="0" />
              <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_ENTRADA" type="xs:dateTime" minOccurs="0" />
              <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
              <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
              <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
              <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
              <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
              <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
REQUER – Número do Requerimento
DESCRICAO – Descrição do Requerimento
DATA_ENTRADA – Data de Entrada do Requerimento
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-400 – Requerimento Inexistente.

4.5. PesqRequerimentos

Pesquisa determinado texto em todos os Requerimentos a que o Munícipe pode aceder.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <PesqReque|imentos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
    </PesqReque~imentos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="PesqRequerimentos">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
              <xs:element name="REQUER" type="xs:string" minOccurs="0" />
              <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_ENTRADA" type="xs:dateTime" minOccurs="0" />
              <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
              <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
              <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
              <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
              <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
              <xs:element name="IDPERFIL" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
REQUER – Número do Requerimento
DESCRICAO – Descrição do Requerimento
DATA_ENTRADA – Data de Entrada do Requerimento
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-403 – Não foram encontrados Requerimentos contendo o texto a pesquisar

Ordenação dos Resultados: IDPERFIL ASC, DATA_ENTRADA DESC

5. Web Service Circulações

O URL que define esta classe de web services é http://ip_servidor_web/WSACentral/wsCirculacoes.asmx.

5.1. LimparCirculacoes

Elimina todos as circulações de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparCirculacoes xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparCirculacoes>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

5.2. LimparCirculacoesProcesso

Elimina todas as circulações de um processo, de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
```

```
    <LimparCirculacoesProcesso xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </LimparCirculacoesProcesso>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Retorno de um DataSet com Códigos de Erro

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

5.3. GetCirculacoes

Retorna um dataset com as circulações de um processo, de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetCirculacoes xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetCirculacoes>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
```

```
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetCirculacoes">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="NUM_CIRC" type="xs:string" minOccurs="0" />
                <xs:element name="SERVICO" type="xs:string" minOccurs="0" />
                <xs:element name="OBSERVACOES" type="xs:string" minOccurs="0" />
                <xs:element name="FUNCIONARIO" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ENVIO" type="xs:dateTime" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:string" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
NUM_CIRC – Número da Circulação
SERVICO – Serviço / Entidade que emitiu a Circulação
OBSERVACOES – Observações
FUNCIONARIO – Nome do Funcionário que emitiu a Circulação –
DATA_ENVIO – Data de Envio da Circulação
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
CONTRIB – Número de Contribuinte do Munícipe
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-303 – Processo sem Circulações.

Ordenação dos Resultados: DATA_ENVIO DESC

5.4. PesqCirculacoes

Pesquisa determinado texto em Circulações.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <PesqCirculacoes xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
    </PesqCirculacoes>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="PesqCirculacoes">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="NUM_CIRC" type="xs:string" minOccurs="0" />
                <xs:element name="SERVICO" type="xs:string" minOccurs="0" />
                <xs:element name="OBSERVACOES" type="xs:string" minOccurs="0" />
                <xs:element name="FUNCIONARIO" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ENVIO" type="xs:dateTime" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:string" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
```

```
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
NUM_CIRC – Número da Circulação
SERVICO – Serviço / Entidade que emitiu a Circulação
OBSERVACOES – Observações
FUNCIONARIO – Nome do Funcionário que emitiu a Circulação
DATA_ENVIO – Data de Envio da Circulação
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
CONTRIB – Número de Contribuinte do Munícipe
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-600 – Texto Inexistente

Ordenação dos Resultados: DATA_ENVIO DESC

6. Web Service Licenças

O URL que define esta classe de web services é http://ip_servidor_web/WSACentral/wsLicencas.asmx.

6.1. LimparLicencas

Elimina todos as Licenças de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparLicencas xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparLicencas>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

6.2. PesqLicencas

Pesquisa determinado texto em Licenças.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
```

```
    <PesqLicencas xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
    </PesqLicencas>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="PesqLicencas">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="N_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="TIP_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_LICENCA" type="xs:dateTime" minOccurs="0" />
                <xs:element name="DESC_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="REQUER" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="PRAZO_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
N_LICENCA – Número da Licença
TIP_LIC – Código do Tipo de Licença
DATA_LICENCA – Data da Licença
DESC_LIC – Descrição da Licença
REQUER – Número do Requerimento
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
PRAZO_LICENCA – Prazo da Licença
PERFIL – Descrição do Perfil do Munícipe perante o Processo

IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-500 – Não foram encontradas Licenças com o texto a pesquisar

Ordenação dos Resultados: DATA_LICENCA DESC

6.3. GetLicencas

Retorna um dataset com as todas as licenças de um de um Munícipe ou todas as licenças de um processo quando este é especificado.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetLicencas xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetLicencas>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo (quando não é especificado usar ‘’)
IDTipoProc – VARCHAR(9) – Tipo do Processo (quando não é especificado usar ‘’)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
```

```
          <xs:element name="GetLicencas">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="N_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="TIP_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_LICENCA" type="xs:dateTime" minOccurs="0" />
                <xs:element name="DESC_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="REQUER" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="PRAZO_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
N_LICENCA – Número da Licença
TIP_LIC – Código do Tipo de Licença
DATA_LICENCA – Data da Licença
DESC_LIC – Descrição da Licença
REQUER – Número do Requerimento
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
PRAZO_LICENCA – Prazo da Licença
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-202 – Munícipe sem Licenças
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-302 – Processo sem Licenças

Ordenação dos Resultados: DATA_LICENCA DESC

7. Web Services Pareceres

O URL que define esta classe de web services é

http://ip_servidor_web/WSACentral/wsPareceres.asmx.

7.1. GetPareceresGlobal

Retorna um dataset com os pareceres de um determinado requerimento ou processo.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetPareceCesGlobal xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
      <IDrequerimento>string</IDrequerimento>
    </GetPareceFesGlobal>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo
IDRequer – VARCHAR(15) – Número do Requerimento. Caso este parâmetro esteja vazio, retorna a lista dos Pareceres por Processo. Se estiver preenchido, retorna a lista dos Pareceres por Requerimento.

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="GetPareceresGlobal">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
              <xs:element name="NUM_PARECER" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_PEDIDO" type="xs:dateTime" minOccurs="0" />
              <xs:element name="DATA_PARECER" type="xs:dateTime" minOccurs="0" />
              <xs:element name="UNID_RESP" type="xs:string" minOccurs="0" />
              <xs:element name="TIPO_PARECER" type="xs:string" minOccurs="0" />
              <xs:element name="RESULTADO" type="xs:string" minOccurs="0" />
              <xs:element name="FICHEIRO" type="xs:string" minOccurs="0" />
              <xs:element name="INT_EXT" type="xs:string" minOccurs="0" />
```

```
            <xs:element name="REQUER" type="xs:string" minOccurs="0" />
            <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
             <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
            <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
            <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
             <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
            <xs:element name="TOOLTIP" type="xs:int" minOccurs="0" />
            <xs:element name="TRANSFERIDO" type="xs:int" minOccurs="0" />
            <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
             <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
NUM_PARECER – Número do Parecer
DATA_PEDIDO – Data de Pedido do Parecer
DATA_PARECER – Data de Emissão do Parecer
UNID_RESP – Unidade Responsável pela Emissão do Parecer
TIPO_PARECER – Código do Tipo de Parecer
RESULTADO – Resultado do Parecer
FICHEIRO – Nome do Ficheiro com o Conteúdo do Parecer
INT_EXT – Indicação de Parecer Interno / Externo
REQUER – Número do Requerimento
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
TOOLTIP – Mensagem a disponibilizar na imagem do Parecer (possivelmente vai ser eliminado)
TRANSFERIDO – Flag que indica se o Conteúdo do Parecer já foi transferido (download)
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-305 – Processo sem pareceres. (utilizado na consulta de Pareceres por Processo)
-400 – Requerimento Inexistente.
-402 – Requerimento sem pareceres

Ordenação dos Resultados: DATA_PARECER DESC

7.2. LimparPareceres

Elimina todos os pareceres de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparPareceres xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparPareceres>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso

-200 – Munícipe inexistente

Este anexo apresenta a descrição WSDL dos Web Services do sistema WSAutarquias.

Todos os schemas de retorno devolvem no dataset a tabela Erros, se o número de retorno for igual ou maior do que 1, então houve sucesso, se for negativo, ocorreu um erro, de acordo com o Web Service correspondente.

```
<xs:element name="Erros">
 <xs:complexType>
   <xs:sequence>
    <xs:element name="Numero" type="xs:string" minOccurs="0" />
    <xs:element name="Descricao" type="xs:string" minOccurs="0" />
   </xs:sequence>
  </xs:complexType>
</xs:element>
```

8. Web Service Munícipe

O URL para chamada a este Web Service é:

http://ip_servidor_web/WSACentral/wsMunicipe.asmx.

Este Web Service define os seguintes Web Methods.

8.1. Autentica

Método Responsável pela autenticação de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <Autentica xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <PIN>string</PIN>
    </Autentica>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe (cifrado com a credencial em uso)
Password – VARCHAR(4) – PIN do Munícipe (cifrado com a credencial em uso)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="Autentica">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="NOME" type="xs:string" minOccurs="0" />
                <xs:element name="ACESSOS" type="xs:int" minOccurs="0" />
                <xs:element name="ACESSOSTOTAIS" type="xs:int" minOccurs="0" />
                <xs:element name="ULTIMOACESSO" type="xs:dateTime" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
```

```
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
NOME – Nome do Munícipe
ACESSOS – Número. de acessos do Munícipe desde a última alteração de PIN efectuada pelo sistema (Nova adesão de Munícipe ou Geração automática do PIN pelo AnswerDesafio). Caso seja 1, deve ser sugerida, ao Munícipe, a alteração do PIN.
ACESSOSTOTAIS – Total de acessos do Munícipe
ULTIMOACESSO – Data / Hora do último acesso efectuado pelo Munícipe

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Password Incorrecta. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-205 – Ultrapassou o número de tentativas erradas. A conta está temporariamente bloqueada. Neste caso, o único acesso passa a ser pelo Método Desafio.

8.2. GetMunicipe

Retorna um dataset com os dados de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetMunicipe xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
    </GetMunicipe>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
```

```
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetMunicipe">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="NOME" type="xs:string" minOccurs="0" />
                <xs:element name="MORADA" type="xs:string" minOccurs="0" />
                <xs:element name="LOCAL" type="xs:string" minOccurs="0" />
                <xs:element name="ESTADO_CIVIL" type="xs:string" minOccurs="0" />
                <xs:element name="FILIACAO_PAI" type="xs:string" minOccurs="0" />
                <xs:element name="FILIACAO_MAE" type="xs:string" minOccurs="0" />
                <xs:element name="PROFISSAO" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_NASC" type="xs:dateTime" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="BI" type="xs:string" minOccurs="0" />
                <xs:element name="BI_DATA" type="xs:dateTime" minOccurs="0" />
                <xs:element name="BI_ARQ" type="xs:string" minOccurs="0" />
                <xs:element name="PASSWORD" type="xs:string" minOccurs="0" />
                <xs:element name="HORA_CARR" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOGINS" type="xs:int" minOccurs="0" />
                <xs:element name="FREG_RESID" type="xs:string" minOccurs="0" />
                <xs:element name="FREG_RESID_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_RESID" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_RESID_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_RESID" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_RESID_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="FREG_NATUR" type="xs:string" minOccurs="0" />
                <xs:element name="FREG_NATUR_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_NATUR" type="xs:string" minOccurs="0" />
                <xs:element name="CONC_NATUR_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_NATUR" type="xs:string" minOccurs="0" />
                <xs:element name="DISTR_NATUR_DESC" type="xs:string" minOccurs="0" />
                <xs:element name="CODIGO_POSTAL" type="xs:string" minOccurs="0" />
                <xs:element name="TELEFONE" type="xs:string" minOccurs="0" />
                <xs:element name="FAX" type="xs:string" minOccurs="0" />
                <xs:element name="EMAIL" type="xs:string" minOccurs="0" />
                <xs:element name="SEXO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPO_USER" type="xs:short" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
NOME – Nome do Munícipe
MORADA – Morada do Munícipe
LOCAL – Localidade
ESTADO_CIVIL – Estado Civil
FILIACAO_PAI – Nome do Pai
FILIACAO_MAE – Nome da Mãe
PROFISSAO – Profissão do Munícipe
DATA_NASC – Data de Nascimento do Munícipe
AUTARQUIA – Código da Autarquia
BI – Número do Bilhete de Identidade
BI_DATA – Data de Emissão do Bilhete de Identidade
BI_ARQ – Arquivo de Emissão do Bilhete de Identidade
PASSWORD – PIN Cifrado
HORA_CARR – Data / Hora em que o Munícipe se Autenticou
LOGINS – Campo mantido para retro-compatibilidade que deverá desaparecer.
FREG_RESID – Código da Freguesia de Residência
FREG_RESID_DESC –Freguesia de Residência

CONC_RESID – Código do Concelho de Residência
CONC_RESID_DESC – Concelho de Residência
DISTR_RESID – Código do Distrito de Residência
DISTR_RESID_DESC – Distrito de Residência
FREG_NATUR – Código da Freguesia da sua Naturalidade
FREG_NATUR_DESC – Freguesia da sua Naturalidade
CONC_NATUR – Código do Concelho da sua Naturalidade
CONC_NATUR_DESC – Concelho da sua Naturalidade
DISTR_NATUR – Código do Distrito da sua Naturalidade
DISTR_NATUR_DESC – Distrito da sua Naturalidade
CODIGO_POSTAL – Código Postal de Residência
TELEFONE – Telefone
FAX – Fax
EMAIL – Email
SEXO – Sexo do Munícipe (F – Feminino / M – Masculino / NULL OU N – Não especificado)
TIPO_USER – Tipo de Utilizador (0 – Utilizador Individual / 1 – Utilizador Empresarial / 2 – Funcionário)

Códigos de Erro:

1 – Sucesso
-100 – Não existe nenhum registo na Autarquia
-101 – Existem dois ou mais registos na Autarquia
-102 – Autarquia não Disponível
-104 – Autarquia em Manutenção
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente

8.3. SetDesafio

Permite definir um desafio que poderá ser usado para gerar uma nova password.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <SetDesafio xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <PIN>string</PIN>
      <Pergunta>string</Pergunta>
      <Resposta>string</Resposta>
    </SetDesafio>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)

PIN – VARCHAR(4) – PIN (cifrado com a credencial em uso)
Pergunta – VARCHAR(200) – Pergunda Desafio (cifrado com a credencial em uso)
Resposta – VARCHAR(200) – Resposta ao Desafio (cifrado com a credencial em uso)

Retorno do DataSet com Códigos de Erro.

Códigos de Erro:

1 – Sucesso
-200 – Munícipe Inexistente
-204 – Password Incorrecta. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-205 – Ultrapassou o número de tentativas erradas. A conta está temporariamente bloqueada. Neste caso, o único acesso passa a ser pelo Método Desafio.

8.4. AskDesafio

Devolve a pergunta Desafio.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <AskDesafio xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <num_contribuinte>string</num_contribuinte>
    </AskDesafio>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe (cifrado com a credencial em uso)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="AskDesafio">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
```

```
              <xs:element name="PERGUNTA" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
 </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
PERGUNTA – Pergunta Desafio

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Acesso negado. Neste caso, o único acesso passa a ser pelo Método Desafio.
-206 – Pergunta Desafio Inexistente.

8.5. AnswearDesafio

Resposta à pergunta Desafio. Em caso de sucesso gera um novo PIN automaticamente e retorna-o.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <AnswerDesafio xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <num_contribuinte>string</num_contribuinte>
      <Resposta>string</Resposta>
    </AnswerDesafio>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe (cifrado com a credencial em uso)
Resposta – VARCHAR(200) – Resposta ao Desafio (cifrado com a credencial em uso)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Lœcale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="AnswerDesafio">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="PIN" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONTRIB – Número de Contribuinte do Munícipe
PIN – Novo PIN gerado pelo Sistema

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Acesso negado.
-207 – Resposta ao desafio errada. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-208 – Ultrapassou o número de tentativas erradas para o Desafio. A conta está temporariamente bloqueada. Neste caso, o utilizador tem mesmo que aguardar as X horas para tentar novamente.

8.6. ChangePassword

Método que permite a alteração do PIN do munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <ChangePassword xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <IDMunicipe>string</IDMunicipe>
      <PINAntigo>string</PINAntigo>
      <PINNovo>string</PINNovo>
    </ChangePassword>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
IDMunicipe – VARCHAR(12) – Número de Munícipe (cifrado com a credencial em uso)
PasswordAntiga – VARCHAR(4) – PIN actualmente utilizado (cifrado com a credencial em uso)
PasswordNovo – VARCHAR(4) – Novo PIN (cifrado com a credencial em uso)

Retorno de DataSet com Códigos de Erro.

Códigos de Erro:

1 – Sucesso
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe Inexistente
-204 – Password Incorrecta. Dispõe de mais X tentativas (X é configurável pela Autarquia e tem o valor por 3 tentativas de login erradas)
-205 – Ultrapassou o número de tentativas erradas. A conta está temporariamente bloqueada. Neste caso, o único acesso passa a ser pelo Método Desafio.

8.7. LimparMunicipe

Limpa toda a informação relacionada com o Munícipe da cache local (Ex. Processo, Perfil, Requerimentos, etc). Este método deverá sempre ser chamado quando um munícipe abandona a sessão web.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparMunicipe xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparMunicipe>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de contribuinte

Retorno de DataSet com Códigos de Erro.

Códigos de Erro:

1 – Sucesso
-200 – Munícipe Inexistente

9. Web Service Processos

O URL que define esta classe de web services é:

http://ip_servidor_web/WSACentral/wsProcessos.asmx.

9.1. LimparProcessos

Elimina todos os processos de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparProcessos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparProcessos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

9.2. GetProcessos

Retorna um dataset com os processos de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetProcessos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
```

```
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
    </GetProcessos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetProcessos">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC_DESCR" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ABERT" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOCAL_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
                <xs:element name="PROPRIETARIO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC_DESCR – Descrição do Tipo de Processo
DATA_ABERT – Data de Abertura do Processo
LOCAL_OBRA – Descrição do Local da Obra
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
PROPRIETARIO – Nome do Proprietário do Processo
TIPOPROC – Código do Tipo de Processo
DESCRICAO – Descrição do Processo
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra

RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-201 – Munícipe sem Processos.

Ordenação dos Resultados: IDPERFIL ASC, DATA_ABERT DESC

9.3. GetProcesso

Retorna um dataset com os dados do processo indicado.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetProcesso xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetProcesso>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
```

```
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetProcesso">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC_DESCR" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ABERT" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOCAL_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
                <xs:element name="PROPRIETARIO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC_DESCR – Descrição do Tipo de Processo
DATA_ABERT – Data de Abertura do Processo
LOCAL_OBRA – Descrição do Local da Obra
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
PROPRIETARIO – Nome do Proprietário do Processo
TIPOPROC – Código do Tipo de Processo
DESCRICAO – Descrição do Processo
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.

9.4. PesqProcessos

Pesquisa determinado texto em Processos.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <PesqProcessos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
    </PesqProcessos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="PesqProcessos">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC_DESCR" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ABERT" type="xs:dateTime" minOccurs="0" />
                <xs:element name="LOCAL_OBRA - Descrição do Local da Obra"
type="xs:string" minOccurs="0" />
                <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
                <xs:element name="PROPRIETARIO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA - Número de Contribuinte do Responsável pela
Responsável pela Direcção Técnica da Obra" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC_DESCR – Descrição do Tipo de Processo
DATA_ABERT – Data de Abertura do Processo
LOCAL_OBRA – Descrição do Local da Obra
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
PROPRIETARIO – Nome do Proprietário do Processo
TIPOPROC – Código do Tipo de Processo
DESCRICAO – Descrição do Processo
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-304 – Não foram encontrados Processos com o texto a pesquisar

Ordenação dos Resultados: IDPERFIL ASC, DATA_ABERT DESC

10. Web Service Requerimentos

O URL que define esta classe de web services é http://ip_servidor_web/WSACentral/wsRequerimentos.asmx.

10.1. LimparRequerimentos

Elimina todos os requerimentos de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparRequerimentos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparRequerimentos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

10.2. LimparRequerimentosProcesso

Elimina todos os requerimentos de processo, de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparRequerimentosProcesso xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
```

```
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </LimparRequerimentosProcesso>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente
-300 – Processo inexistente

10.3. GetRequerimentos

Retorna um dataset com os requerimentos de um processo, de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetRequerimentos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetRequerimentos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
```

```
        <xs:element name="GetRequerimentos">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
              <xs:element name="REQUER" type="xs:string" minOccurs="0" />
              <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_ENTRADA" type="xs:dateTime" minOccurs="0" />
              <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
              <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
              <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
              <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
              <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
              <xs:element name="IDPERFIL" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
REQUER – Número do Requerimento
DESCRICAO – Descrição do Requerimento
DATA_ENTRADA – Data de Entrada do Requerimento
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao processo.
-300 – Processo inexistente.
-301 – Processo sem requerimentos

Ordenação dos Resultados: IDPERFIL ASC, DATA_ENTRADA DESC

10.4. GetRequerimento

Retorna um dataset com os dados do requerimento indicado.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetRequerimento xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
      <IDrequerimento>string</IDrequerimento>
    </GetRequerimento>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo
IDRequer – VARCHAR(15) - Número do Requerimento

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="GetRequerimento">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
               <xs:element name="REQUER" type="xs:string" minOccurs="0" />
              <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_ENTRADA" type="xs:dateTime" minOccurs="0" />
              <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
              <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
              <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
              <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
              <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
              <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
```

```
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
REQUER – Número do Requerimento
DESCRICAO – Descrição do Requerimento
DATA_ENTRADA – Data de Entrada do Requerimento
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-400 – Requerimento Inexistente.

10.5. PesqRequerimentos

Pesquisa determinado texto em todos os Requerimentos a que o Munícipe pode aceder.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <PesqReque|imentos xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
```

```
    </PesqReque~imentos>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="PesqRequerimentos">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
              <xs:element name="REQUER" type="xs:string" minOccurs="0" />
              <xs:element name="DESCRICAO" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_ENTRADA" type="xs:dateTime" minOccurs="0" />
              <xs:element name="COM_DETALHE" type="xs:string" minOccurs="0" />
              <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
              <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
              <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
              <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
              <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
              <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
              <xs:element name="IDPERFIL" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
REQUER – Número do Requerimento
DESCRICAO – Descrição do Requerimento
DATA_ENTRADA – Data de Entrada do Requerimento
COM_DETALHE – Flag que indica se o Processo tem Detalhe (Requerimentos, Circulações, Pareceres, Notificações) ou não.
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo

IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-403 – Não foram encontrados Requerimentos contendo o texto a pesquisar

Ordenação dos Resultados: IDPERFIL ASC, DATA_ENTRADA DESC

11. Web Service Circulações

O URL que define esta classe de web services é http://ip_servidor_web/WSACentral/wsCirculacoes.asmx.

11.1. LimparCirculacoes

Elimina todos as circulações de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparCirculacoes xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparCirculacoes>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

11.2. LimparCirculacoesProcesso

Elimina todas as circulações de um processo, de um Munícipe na cache.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
```

```
    <LimparCirculacoesProcesso xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </LimparCirculacoesProcesso>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Retorno de um DataSet com Códigos de Erro

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

11.3. GetCirculacoes

Retorna um dataset com as circulações de um processo, de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetCirculacoes xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetCirculacoes>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
```

```
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="GetCirculacoes">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="NUM_CIRC" type="xs:string" minOccurs="0" />
                <xs:element name="SERVICO" type="xs:string" minOccurs="0" />
                <xs:element name="OBSERVACOES" type="xs:string" minOccurs="0" />
                <xs:element name="FUNCIONARIO" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ENVIO" type="xs:dateTime" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:string" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
NUM_CIRC – Número da Circulação
SERVICO – Serviço / Entidade que emitiu a Circulação
OBSERVACOES – Observações
FUNCIONARIO – Nome do Funcionário que emitiu a Circulação –
DATA_ENVIO – Data de Envio da Circulação
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
CONTRIB – Número de Contribuinte do Munícipe
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-303 – Processo sem Circulações.

Ordenação dos Resultados: DATA_ENVIO DESC

11.4. PesqCirculacoes

Pesquisa determinado texto em Circulações.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <PesqCirculacoes xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
    </PesqCirculacoes>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="PesqCirculacoes">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="NUM_CIRC" type="xs:string" minOccurs="0" />
                <xs:element name="SERVICO" type="xs:string" minOccurs="0" />
                <xs:element name="OBSERVACOES" type="xs:string" minOccurs="0" />
                <xs:element name="FUNCIONARIO" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_ENVIO" type="xs:dateTime" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:string" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_OBRA" type="xs:string" minOccurs="0" />
                <xs:element name="RESP_PROJ" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
```

```
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
NUM_CIRC – Número da Circulação
SERVICO – Serviço / Entidade que emitiu a Circulação
OBSERVACOES – Observações
FUNCIONARIO – Nome do Funcionário que emitiu a Circulação
DATA_ENVIO – Data de Envio da Circulação
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
CONTRIB – Número de Contribuinte do Munícipe
RESP_OBRA – Número de Contribuinte do Responsável pela Responsável pela Direcção Técnica da Obra
RESP_PROJ – Número de Contribuinte do Responsável pelo Projecto de Arquitectura
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-600 – Texto Inexistente

Ordenação dos Resultados: DATA_ENVIO DESC

12. Web Service Licenças

O URL que define esta classe de web services é http://ip_servidor_web/WSACentral/wsLicencas.asmx.

12.1. LimparLicencas

Elimina todos as Licenças de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparLicencas xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparLicencas>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

12.2. PesqLicencas

Pesquisa determinado texto em Licenças.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
```

```
    <PesqLicencas xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <texto>string</texto>
    </PesqLicencas>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
Texto – VARCHAR(200) – Texto a pesquisar

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="PesqLicencas">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="N_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="TIP_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_LICENCA" type="xs:dateTime" minOccurs="0" />
                <xs:element name="DESC_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="REQUER" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="PRAZO_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
N_LICENCA – Número da Licença
TIP_LIC – Código do Tipo de Licença
DATA_LICENCA – Data da Licença
DESC_LIC – Descrição da Licença
REQUER – Número do Requerimento
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
PRAZO_LICENCA – Prazo da Licença
PERFIL – Descrição do Perfil do Munícipe perante o Processo

IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-500 – Não foram encontradas Licenças com o texto a pesquisar

Ordenação dos Resultados: DATA_LICENCA DESC

12.3. GetLicencas

Retorna um dataset com as todas as licenças de um de um Munícipe ou todas as licenças de um processo quando este é especificado.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetLicencas xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
    </GetLicencas>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo (quando não é especificado usar ‘’)
IDTipoProc – VARCHAR(9) – Tipo do Processo (quando não é especificado usar ‘’)

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://ws.shortcut.pt/ws.WSAutarquiasRemoto/">
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
```

```
          <xs:element name="GetLicencas">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
                <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
                <xs:element name="N_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="TIP_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="DATA_LICENCA" type="xs:dateTime" minOccurs="0" />
                <xs:element name="DESC_LIC" type="xs:string" minOccurs="0" />
                <xs:element name="REQUER" type="xs:string" minOccurs="0" />
                <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
                <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
                <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
                <xs:element name="PRAZO_LICENCA" type="xs:string" minOccurs="0" />
                <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
                <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</DataSet>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
CONTRIB – Número de Contribuinte do Munícipe
N_LICENCA – Número da Licença
TIP_LIC – Código do Tipo de Licença
DATA_LICENCA – Data da Licença
DESC_LIC – Descrição da Licença
REQUER – Número do Requerimento
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
PRAZO_LICENCA – Prazo da Licença
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-202 – Munícipe sem Licenças
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-302 – Processo sem Licenças

Ordenação dos Resultados: DATA_LICENCA DESC

13. Web Services Pareceres

O URL que define esta classe de web services é

http://ip_servidor_web/WSACentral/wsPareceres.asmx.

13.1. GetPareceresGlobal

Retorna um dataset com os pareceres de um determinado requerimento ou processo.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <GetPareceCesGlobal xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <cod_autarquia>string</cod_autarquia>
      <num_contribuinte>string</num_contribuinte>
      <tcProcesso>string</tcProcesso>
      <tcTipo>string</tcTipo>
      <IDrequerimento>string</IDrequerimento>
    </GetPareceFesGlobal>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Cod_autarquia – VARCHAR(11) – Código da autarquia
Num_contribuinte – VARCHAR(11) – Número de Contribuinte do Munícipe
IDProcesso – VARCHAR(10) – Número do Processo
IDTipoProc – VARCHAR(9) – Tipo do Processo
IDRequer – VARCHAR(15) – Número do Requerimento. Caso este parâmetro esteja vazio, retorna a lista dos Pareceres por Processo. Se estiver preenchido, retorna a lista dos Pareceres por Requerimento.

Envelope de Retorno:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="pt-PT">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="GetPareceresGlobal">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CONSULTOR" type="xs:string" minOccurs="0" />
              <xs:element name="NUM_PARECER" type="xs:string" minOccurs="0" />
              <xs:element name="DATA_PEDIDO" type="xs:dateTime" minOccurs="0" />
              <xs:element name="DATA_PARECER" type="xs:dateTime" minOccurs="0" />
              <xs:element name="UNID_RESP" type="xs:string" minOccurs="0" />
              <xs:element name="TIPO_PARECER" type="xs:string" minOccurs="0" />
              <xs:element name="RESULTADO" type="xs:string" minOccurs="0" />
              <xs:element name="FICHEIRO" type="xs:string" minOccurs="0" />
              <xs:element name="INT_EXT" type="xs:string" minOccurs="0" />
```

```
            <xs:element name="REQUER" type="xs:string" minOccurs="0" />
            <xs:element name="CONTRIB" type="xs:string" minOccurs="0" />
             <xs:element name="AUTARQUIA" type="xs:int" minOccurs="0" />
            <xs:element name="PROCESSO" type="xs:string" minOccurs="0" />
            <xs:element name="TIPOPROC" type="xs:string" minOccurs="0" />
             <xs:element name="TIPOREQ" type="xs:string" minOccurs="0" />
            <xs:element name="TOOLTIP" type="xs:int" minOccurs="0" />
            <xs:element name="TRANSFERIDO" type="xs:int" minOccurs="0" />
            <xs:element name="PERFIL" type="xs:string" minOccurs="0" />
             <xs:element name="IDPERFIL" type="xs:int" minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Descrição dos Campos Devolvidos:

CONSULTOR – Número de Contribuinte do Munícipe que está a consultar o Processo (normalmente é igual ao campo CONTRIB, excepto no caso em que hajam Utilizadores do tipo FUNCIONÁRIO)
NUM_PARECER – Número do Parecer
DATA_PEDIDO – Data de Pedido do Parecer
DATA_PARECER – Data de Emissão do Parecer
UNID_RESP – Unidade Responsável pela Emissão do Parecer
TIPO_PARECER – Código do Tipo de Parecer
RESULTADO – Resultado do Parecer
FICHEIRO – Nome do Ficheiro com o Conteúdo do Parecer
INT_EXT – Indicação de Parecer Interno / Externo
REQUER – Número do Requerimento
CONTRIB – Número de Contribuinte do Munícipe
AUTARQUIA – Código da Autarquia
PROCESSO – Número do Processo
TIPOPROC – Código do Tipo de Processo
TIPOREQ – Código do Tipo de Requerimento
TOOLTIP – Mensagem a disponibilizar na imagem do Parecer (possivelmente vai ser eliminado)
TRANSFERIDO – Flag que indica se o Conteúdo do Parecer já foi transferido (download)
PERFIL – Descrição do Perfil do Munícipe perante o Processo
IDPERFIL – Código do Perfil do Munícipe perante o Processo (1 – Titular do Processo / 2 – Responsável pela Direcção Técnica / 3 – Responsável pelo Projecto de Arquitectura / 4 – Outro Técnico Responsável / 5 – Outro Requerente / 6 – Funcionário)

Códigos de Erro:

1 – Dados Devolvidos com sucesso.
-100 – Não existe nenhum registo na Autarquia.
-101 – Existem dois ou mais registos na Autarquia.
-102 – Autarquia não Disponível.
-104 – Autarquia em Manutenção.
-105 – Erro de Comunicações com a Autarquia – WebService não encontrado
-200 – Munícipe inexistente.
-203 – Munícipe sem permissões para aceder ao Processo.
-300 – Processo inexistente.
-305 – Processo sem pareceres. (utilizado na consulta de Pareceres por Processo)
-400 – Requerimento Inexistente.
-402 – Requerimento sem pareceres

Ordenação dos Resultados: DATA_PARECER DESC

13.2. LimparPareceres

Elimina todos os pareceres de um Munícipe.

Envelope de Entrada:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SHWSClientCredentials xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <SHWSCredentialId>string</SHWSCredentialId>
    </SHWSClientCredentials>
  </soap:Header>
  <soap:Body>
    <LimparPareceres xmlns="http://ws.shortcut.pt/ws.WSAutarquiasCentral/">
      <num_contribuinte>string</num_contribuinte>
    </LimparPareceres>
  </soap:Body>
</soap:Envelope>
```

Parâmetros de Entrada:

Num_contribuinte – VARCHAR(11) – Número de Contribuinte

Retorno de um DataSet com Códigos de Erro.

Códigos de Erro:

1 – Dados eliminados com sucesso
-200 – Munícipe inexistente

Anexo II

Código que implementa a integração, o sistema de segurança no cliente e a utilização de SSL

Neste anexo apresentam-se os procedimentos para estabelecer, na aplicação cliente TRAS-OS-MONTES.NET, o acesso aos Web Services WSAutarquias, usando a componente de segurança. O código apresentado usa a linguagem VB.NET, mas poderia ser utilizada qualquer linguagem de desenvolvimento suportada pela plataforma .NET 1.1 ou superior.

Começa-se por estabelecer três referências na aplicação cliente, a primeira é uma *Web Reference* para o Web Service de autenticação disponível no endereço `http://ws.shortcut.pt/WSACentral/wsAAuthentication.asmx`, a qual designamos por WebAutenticacao, este é o nome com que o serviço remoto de autenticação é conhecido na aplicação cliente.

A segunda referência é para o *Namespace* da *.Net Framework* denominado `System.Web.Services.Protocols`, esta torna possível utilizar os *Web Methods* localizados no servidor remoto, através dos protocolos padrão da *Internet*, como se fossem funções locais.

A terceira referência é para a componente SHSecurityClient.dll instalada localmente, esta vai permitir usar as suas classes que estão reunidas sobre o Namespace `Shortcut.Security`.

```
Imports wsAutarquiasTMD.WebAutenticacao
Imports System.Web.Services.Protocols
Imports Shortcut.Security
```

Tal como ilustrado de seguida, a implementação na aplicação começa por criar três objectos. `Auth` e `Authcrt` são usados para efectuar a autenticação no servidor remoto usando uma credencial, enquanto `SecurityClient` é usado para efectuar a encriptação de dados localmente.

```
Public Auth As WebAutenticacao.WSAAuthentication
Public Authcrt As  WebAutenticacao.SHWSClientCredentials
Public SecurityClient As Shortcut.Security.SHCertifyedClient
```

Definiu-se um método para inicializar os objectos `Auth` e `Authcrt` antes de se usarem, o método designou-se `initAuth` e é apresentado a seguir.

```
Private Sub initAuth()
      Authcrt = New WebAutenticacao.SHWSClientCredentials
      Auth = New WebAutenticacao.WSAAuthentication
End Sub
```

De seguida implementou-se a função `obterCredencial`, esta solicita uma credencial ao Web Service de autenticação invocando o *Web Method* `getValidKey`.

```
Private Function obterCredencial() As String
      Try
            initAuth()
            obterCredencial = Auth.getValidKey()
      Catch se As SoapException
            Throw se
      Catch ex As Exception
            Throw ex
      End Try
End Function
```

A credencial obtida é usada na função `Autenticar` para cifrar os dados de *login* e *password* usando o método `getEncrypt da classe SHCertifyedClient` no Namespace referido `Shortcut.Security`. Os dados cifrados enviam-se ao servidor remoto através do no método `Authenticate` para pedir uma credencial autenticada.

```
Private Function Autenticar(ByVal user As UserInfo) As String

      Dim log As New Log(user)
      Try
            initAuth()
            Authcrt.SHWSCredentialId = Auth.getValidKey()
```

```
            Auth.SHWSClientCredentialsValue = Authcrt

            Dim user_encrypt As String
            Dim pass_encrypt As String

            pass_encrypt = SecurityClient.getEncrypt(user.Login, _
                                        Authcrt.SHWSCredentialId)
            user_encrypt = SecurityClient.getEncrypt(user.Pwd, _
                                        Authcrt.SHWSCredentialId)

            If (Auth.Authenticate(user_encrypt, pass_encrypt)) Then
                log.AddMessage("Authentication OK.")
                Autenticar = Authcrt.SHWSCredentialId
            Else
                log.AddMessage("Authentication NOT OK.")
                Autenticar = False.ToString()
            End If

      Catch se As SoapException
            Throw New Exception("SoapException: " & se.Message)
            log.AddMessage("SoapException. Authentication NOT OK.")

      Catch ex As Exception
            Throw ex

End Try

End Function
```

Se esta autenticação for efectuada com sucesso podemos usar a credencial autenticada para aceder aos restantes Web Services.

A credencial expira quando um dos Web Services envia a seguinte mensagem: "A SHWSCredencialId usada não é válida, peça uma nova!" e o *faultCode* `SoapException.VersionMismatchFaultCode`, então é necessário efectuar todos os passos descritos anteriormente, voltando a solicitar uma credecial válida.

Há semelhança de todo o código desenvolvido, utilizam-se os blocos Try Catch para que todas as excepções imprevistas que ocorram em run-time possam ser capturadas e

tratadas, além disso, incorporaram-se também funções de *logging* para registar as excepções que ocorrem.

Este tipo de metodologia pode ser implementada sobre SSL, descreve-se de seguida, como configurar e codificar esta lógica para usar sobre HTTPS.

Implementação sobre SSL

A implementação sobre SSL requer que seja feito o *override* de uma função de segurança e ainda o registo desta na aplicação cliente.

A classe que efectua override de uma função de segurança é apresentada a seguir:

```
Imports System.Net
Imports Shortcut.Security
Imports System.Security.Cryptography.X509Certificates

Public Class MyCertificateValidation

    Implements System.Net.ICertificatePolicy

    Public Function CheckValidationResult(ByVal sp As ServicePoint, _
                                          ByVal cert As X509Certificate, _
                                          ByVal req As WebRequest, _
                                          ByVal prob As Integer) _
        As Boolean Implements ICertificatePolicy.CheckValidationResult

        Return True

    End Function

End Class
```

Na parte em que se vai consumir os Web Services usando SSL, regista-se esta classe de segurança do seguinte modo:

```
ServicePointManager.CertificatePolicy = New MyCertificateValidation
```

Anexo III

Código que implementa a integração de modo síncrono e assíncrono

Neste anexo exemplifica-se a invocação aos Web Services de duas formas distintas: chamadas síncronas e chamadas assíncronas.

A função `ConsultarProcessos` apresentada nos dois exemplos que se seguem invoca o *Web Method* `GetProcessos` do Web Service Processos. Este método devolve um objecto do tipo *DataSet* com a lista de processos solicitada e com esta informação cria uma tabela de HTML para apresentar numa página web.

1. Exemplo com chamada síncrona

O seguinte código implementa um processo síncrono, significa que logo após a invocação é esperada uma resposta, e o processo fica bloqueado à espera do retorno para prosseguir.

```
Public Function ConsultarProcessos(ByVal sCredencial As String, _
                                   ByVal sAutarquia As String, _
                                   ByVal sNIF As String) As DataSet

    Dim ds As New DataSet

    Try

      ' Cria objetos para chamar WebService
      initAuth()
      Authcrt.SHWSCredentialId = sCredencial
      wsProcessos = New wsAutarquiasTMD.wsProcessos.wsProcessos
      wsProcessos.SHWSClientCredentialsValue = Authcrt

      ' Chama Web Method GetProcessos
      ds = wsProcessos.GetProcessos(sAutarquia, sNIF)

      Catch se As SoapException
            Throw se

        Catch ex As Exception
            Throw ex

      End Try

      ConsultarProcessos = ds

End Sub
```

2. Exemplo com chamada assíncrona

A implementação seguinte, codifica uma chamada assíncrona, em que o cliente que efectua a chamada não espera o retorno antes de continuar, mas quando a resposta estiver disponível é chamada o procedimento `ReturnProcessos`. Note-se no código que a chamada assíncrona usa os métodos começados por `Begin` e `End`, o primeiro aceita como parâmetros de entrada extra o método a ser chamado quando a resposta estiver disponível.

```
Public Sub ConsultarProcessos(ByVal sCredencial As String, _
                              ByVal sAutarquia As String, _
                              ByVal sNIF As String)

      Dim ds As New DataSet

      Try
         ' Cria objetos para chamar WebService
         initAuth()
         Authcrt.SHWSCredentialId = sCredencial
         wsProcessos = New wsAutarquiasTMD.wsProcessos.wsProcessos
         wsProcessos.SHWSClientCredentialsValue = Authcrt

         wsProcessos.BeginGetProcessos(sAutarquia, sNIF, _
            New AsyncCallback(AddressOf ReturnProcessos), wsProcessos)

      Catch se As SoapException
            Throw se

      Catch ex As Exception
            Throw ex
      End Try

    End Sub


Sub ReturnProcessos(ByVal A As IAsyncResult)

      Try
            Dim ds As New DataSet
            wsProcessos = A.AsyncState
            ds = wsProcessos.EndGetProcessos(A)

      Catch se As SoapException
            Throw se

      Catch ex As Exception
            Throw ex
      End Try

End Sub
```